

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ**

DEPARTMENT OF BIOMEDICAL ENGINEERING

**VYUŽITÍ METOD TVAROVÉ ANALÝZY PRO KLASIFIKACI  
OBJEKTŮ V MEDICÍNSKÝCH OBRAZECH**

UTILISATION OF SHAPE ANALYSIS METHODS FOR OBJECT CLASSIFICATION IN MEDICAL IMAGES

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Jiří Karela**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Jiří Chmelík**

**BRNO 2020**

# Bakalářská práce

bakalářský studijní obor **Biomedicínská technika a bioinformatika**

Ústav biomedicínského inženýrství

**Student:** Jiří Karel

**ID:** 203668

**Ročník:** 3

**Akademický rok:** 2019/20

## NÁZEV TÉMATU:

**Využití metod tvarové analýzy pro klasifikaci objektů v medicínských obrazech**

## POKYNY PRO VYPRACOVÁNÍ:

1) Seznamte se s problematikou tvarové analýzy obrazových dat a proveďte literární rešerši dostupných metod. 2) Navrhněte vhodné metody pro určení tvarových vlastností objektů v obrazových datech. 3) Navržené metody implementujte ve zvoleném programovacím jazyce a algoritmy otestujte na simulovaných datech. Vyberte pro tato data relevantní tvarové příznaky z hlediska klasifikace objektů. 4) Navrhněte, implementujte a naučte jednoduchý klasifikátor a otestujte jej na simulovaných datech. 5) Navržené algoritmy dále otestujte na reálných medicínských datech. 6) Dosažené výsledky vyhodnotte a diskutujte.

## DOPORUČENÁ LITERATURA:

[1] COSTA, L. da Fontoura a R. M. CESAR. Shape analysis and classification: theory and practice. Boca Raton, FL: CRC Press, 2001. ISBN 978-084-9334-931.

[2] NEAL, F. B. a J. C. RUSS. Measuring shape. 1. Boca Raton: CRC Press, 2012. ISBN 978-143-9855-980.

[3] DHARA, A. K., S. MUKHOPADHYAY, A. DUTTA, M. GARG a N. KHANDELWAL. A Combination of Shape and Texture Features for Classification of Pulmonary Nodules in Lung CT Images. Journal of Digital Imaging. 2016, 29(4), 466-475. DOI: 10.1007/s10278-015-9857-6. ISSN 0897-1889.

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 5.6.2020

**Vedoucí práce:** Ing. Jiří Chmelík

**prof. Ing. Ivo Provazník, Ph.D.**  
předseda oborové rady

## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Tato bakalářská práce se zabývá problematikou tvarové analýzy. Popisuje některé postupy a metody, které s touto analýzou souvisí. Práce je rozdělena na teoretickou část, praktickou část a závěr.

V teoretické části jsou popsány do většího detailu některé metody, s pomocí kterých poté byla řešena část praktická. Také je zde ale popsána i další teorie, která souvisí s tématem. V praktické části se poté navazuje na danou teorii a je řešen problém tvarové analýzy díky znalostem v teorii získaných. Algoritmus je otestován na medicínských datech z CT obratel.

Závěr slouží jako shrnutí a zhodnocení řešení tvarové analýzy. Také slouží jako úvaha nad realizací naší metody, tedy jak by se mohlo naše řešení a výsledek zlepšit.

## KLÍČOVÁ SLOVA

tvar, obraz, obrys, deskriptor, výpočetní tomografie

## ABSTRACT

Bachelor thesis deals with problems of shape analysis. It describes some procedures and methods related to this kind of analysis. The thesis is divided into theoretical part, practical part and conclusion.

In the theoretical part we describe in greater detail some methods, with the help of which the practical part was solved. But other theories related to the topic are also described. The practical part then follows the given theory and solves the problem of shape analysis due to the knowledge gained in the theory. The algorithm is tested on medical data from CT of vertebrae.

The conclusion serves as a summary and evaluation of the shape analysis solution. It also serves as a reflection on the realization of our method, ie how our solution and result could be improved.

## KEYWORDS

shape, image, contour, descriptor, computed tomography

KARELA, Jiří. *Metoda tvarové analýzy obrazových dat*. Brno, Rok, 73 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav Biomedicínského inženýrství. Vedoucí práce: Ing. Jiří Chmelík

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Metoda tvarové analýzy obrazových dat“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Jiřímu Chmelíkovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora

# Obsah

Úvod	11
<b>1 Teoretická část</b>	<b>12</b>
1.1 Morfologické zpracování obrazových dat . . . . .	12
1.1.1 Základní morfologické operace . . . . .	12
1.1.2 Skelet . . . . .	15
1.1.3 Kontura . . . . .	15
1.2 Segmentace obrazu . . . . .	17
1.2.1 Detekce hran . . . . .	18
1.2.2 Segmentace založená na regionech . . . . .	18
1.2.3 Obrysové trasování . . . . .	19
1.3 Reprezentace a popis obrazu . . . . .	20
1.3.1 Reprezentace tvaru na základě kontur . . . . .	21
1.3.2 Globální metody . . . . .	21
1.3.3 Strukturní metody . . . . .	26
1.3.4 Geometrické deskriptory . . . . .	29
1.4 Klasifikace obrazových dat . . . . .	34
1.4.1 Vícevrstvá dopředná neuronová síť . . . . .	34
1.4.2 Korelační analýza . . . . .	36
1.5 Aplikace tvarové analýzy . . . . .	37
1.5.1 Tvarová analýza mozkových komor s pomocí SPHARM . . . . .	38
1.5.2 Detekce plicních uzlin v CT obrazech hrudníku s pomocí tvarové analýzy . . . . .	38
1.5.3 Tvarová analýza mamografických kalcifikací . . . . .	40
1.5.4 Modelová analýza lokálního tvaru pro detekci lézí v CT skenech	41
<b>2 Praktická část</b>	<b>43</b>
2.1 Aplikace tvarové analýzy na základní 2D obrazy . . . . .	43
2.1.1 Použitá obrazová data . . . . .	43
2.1.2 Předzpracování obrazových dat . . . . .	45
2.1.3 Aplikace tvarových deskriptorů . . . . .	47
2.1.4 Klasifikace objektů . . . . .	52
2.1.5 Diskuze výsledků klasifikace geometrických objektů . . . . .	56
2.2 Aplikace tvarové analýzy na medicínská data . . . . .	60
2.2.1 Obrazová CT data . . . . .	60
2.2.2 Klasifikace obratlů . . . . .	61
2.2.3 Diskuze výsledků klasifikace obratlů . . . . .	67

<b>3 Závěr</b>	<b>70</b>
<b>Seznam zkratk</b>	<b>71</b>
<b>Literatura</b>	<b>72</b>



# Seznam obrázků

1.1	Dvě isotropické binární masky. Vlevo 4-spojené sousední pixely. Vpravo 8-spojené sousední pixely. Zdroj: [12]	13
1.2	Přehled metod segmentace obrazu. Zdroj: [7]	18
1.3	Klasifikace reprezentace a popisu tvaru. Zdroj: [28]	20
1.4	Excentricita a kruhovitost tvaru. Zdroj: [28]	22
1.5	Tvar jablka a jeho centroidní vzdálenost. Zdroj: [28]	22
1.6	(a) Tvar jablka;(b) obrys jablka (a);(c) distanční funkce centroidu jablka (a). Zdroj: [27]	23
1.7	(a) Tvar bez obrysu;(b),(c) dva tvary se stejným obrysem, ale jiným vnitřním obsahem. Zdroj: [27]	24
1.8	Segment obrysu a reprezentace jako 1-D funkce. Zdroj: [10]	26
1.9	Řetězové vektory: (a) původní popis obrysu; (b) normalizovaný popis obrysu. Zdroj: [27]	29
1.10	Standardní MBR a její škálovaná verze. Zdroj: [19]	31
1.11	Přilnutí robustního MBR k různým tvarům. Zdroj: [19]	32
1.12	Přehled klasifikačních metod. Zdroj: [11]	34
1.13	Vícevrstevná dopředná neuronová síť. Vlevo je počet vstupních proměnných. Následuje skrytá vrstva s deseti neurony. Na výstupu jsou tři klasifikační třídy. Zdroj: [15]	36
1.14	Segmentace plic s pomocí metody neostrého prahování. (a) Původní obraz z CT plic; (b) obraz po prahování; (c) segmentované konkávní regiony (připojená uzlina a další tkáně); (d) výsledný segmentovaný obraz na základě (b) a (c). Zdroj: [26]	40
1.15	Tři příklady struktur, které nebyly zaznamenány radiologem, ale nezávisle na tom klasifikovány jako uzliny dalšími dvěma radiology. Tyto abnormality byly úspěšně detekovány algoritmem navrženým v článku. Zdroj: [16]	42
2.1	Čtverec vytvořený v programovacím prostředí	44
2.2	Trojúhelník vytvořený v programovacím prostředí	44
2.3	Vygenerovaný obraz elipsy.	46
2.4	Získaný obrys vygenerované elipsy s pomocí Cannyho detektoru.	46
2.5	Obraz s mincemi po aplikaci obrysového trasování. Červenou barvou je vyznačen obrys mincí.	47
2.6	Vygenerovaný obraz kruhu.	48
2.7	Vzdálenost bodů obrysu od centroidu u kruhu	50
2.8	Vzdálenost bodů obrysu od centroidu u čtverce	51
2.9	Vzdálenost bodů obrysu od centroidu u trojúhelníku	51

2.10	Binární obraz s několika klasifikovanými tvary najednou. . . . .	53
2.11	Vygenerovaný kruh obsahující náhodný šum . . . . .	54
2.12	Filtrovaný obraz od náhodného šumu s pomocí gaussova filtru . . . .	55
2.13	Jeden řez obratle Th1 . . . . .	60
2.14	Jeden řez obratle L1 . . . . .	61
2.15	Vývojový diagram algoritmu pro klasifikaci obratlů . . . . .	62
2.16	Vyplněný obratel Th1 po mediánové filtraci . . . . .	63
2.17	Páteční kanál Th1 po mediánové filtraci . . . . .	63
2.18	Nalezená kontura obratle Th1 po předešlých úpravách . . . . .	64
2.19	Vzdálenost bodů obrysu od centroidu u thorakálního obratle Th1 (modrou barvou) a lumbálního obratle L5 (červenou barvou) . . . . .	65

# Seznam tabulek

2.1	Tabulka s vypočítanými hodnotami kruhovitosti vygenerovaných tvarů s rozlišením 200px . . . . .	49
2.2	Tabulka s nejnižší hodnotou klasifikace geometrických objektů pro různý počet vrstev a neuronů . . . . .	58
2.3	Tabulka s vypočítanými hodnotami klasifikace geometrických objektů	58
2.4	Tabulka s výsledky klasifikace geometrických objektů . . . . .	58
2.5	Tabulka s výsledky senzitivity a PPH klasifikace . . . . .	59
2.6	Tabulka s výsledky korelační analýzy. Vyjadřuje korelaci mezi jednotlivými příznaky dle zadaných hypotéz. . . . .	66
2.7	Tabulka s nejnižší hodnotou klasifikace obratlů pro různý počet vrstev a neuronů . . . . .	68
2.8	Tabulka s vypočítanými hodnotami klasifikace obratlů . . . . .	69
2.9	Tabulka s výsledky klasifikace obratlů . . . . .	69
2.10	Tabulka s výsledky senzitivity a PPH klasifikace obratlů . . . . .	69

# Úvod

Tato práce se věnuje oblasti tvarové analýzy obrazových dat. Cílem tvarové analýzy je popis objektů a tvarů, které se nachází v obrazových datech. V obrazech nalézáme velké množství různých parametrů, na jejichž základě můžeme objekty v nich obsažené klasifikovat. Různé tvary se dají popsat s pomocí tvarových deskriptorů, které jsou založené na různých metodách. Tvar je mocným prostředkem pro popis a rozlišování objektů a je velmi hojně využíván k dosažení výsledků v mnoha oblastech biomedicíny, počítačového vidění a dalších.

V této práci budou tedy představeny a vysvětleny některé metody tvarové analýzy. Zejména se budeme zabývat morfologickým zpracováním, reprezentací a popisem obrazu a zmíníme také segmentaci obrazu. Následně některé z metod popsaných v teorii budou aplikovány na konkrétní obrazová data, která jsme vygenerovali, a budeme sledovat, jaký mají na výsledek analýzy a klasifikace vliv.

Nejdříve budou metody testovány na jednoduchých obrazových datech, obsahujících například běžné geometrické tvary, jako je čtverec, kruh, trojúhelník, elipsa a další, která jsme získali generováním v programovacím prostředí o různých parametrech nebo z externích zdrojů, abychom zjistili funkčnost navržených metod řešení. Hlavně nás bude zajímat schopnost našeho algoritmu klasifikovat správně tvary objektů v obrazech. Poté budeme pracovat s pomocí navrženého algoritmu také s obrazy z lékařské databáze, konkrétně s obrazovými daty z počítačové tomografie. Algoritmus bude pro tuto práci mírně upraven. Budeme se zabývat klasifikací obratlů do kategorií thorakální (hrudní) a lumbální (bederní) na základě jejich tvarových charakteristik.

# 1 Teoretická část

Teoretická část je rozdělena do podkapitol Segmentace obrazových dat, Morfologické zpracování obrazových dat, Reprezentace a popis obrazových dat, Klasifikace obrazových dat a Aplikace tvarové analýzy. Tyto podkapitoly popisují postupy a metody, které slouží ke zpracování obrazových dat od jejich analýzy až po následnou klasifikaci, a také využití tvarové analýzy v praxi. Tato část práce slouží jako seznámení s problematikou tvarové analýzy a předloha pro následné řešení na konkrétních příkladech.

## 1.1 Morfologické zpracování obrazových dat

Morfologie běžně označuje oblast biologie, která se zabývá konkrétními formami či strukturou zvířat a rostlin. V tomto kontextu používáme slovo morfologie ve smyslu matematické morfologie, kterou využíváme k získání komponentů v obraze, které jsou užitečné v reprezentaci a popisu tvaru regionů, mezi které můžeme zařadit obrys, skelet a další. V této kapitole se také budeme zabývat morfologickými technikami, které slouží k předzpracování či zpracování obrazu, mezi které patří například morfologické filtrování. [10]

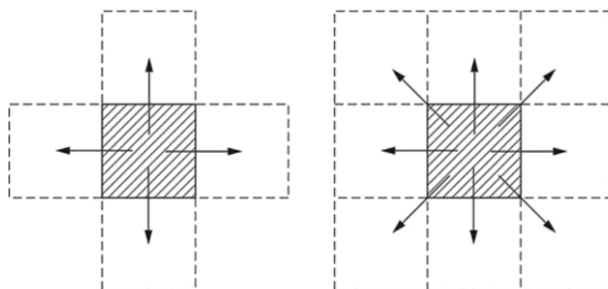
V následujících sekcích se budeme věnovat několika důležitým konceptům, které patří do matematické morfologie. Mnoho z těchto konceptů, které uvedeme, můžeme formulovat v řádu  $n$ -dimensionálního euclidovského prostoru  $E_N$ . Ovšem nejdříve se podíváme na binární obrazy, jejichž komponenty jsou elementy  $Z_2$ . [10]

Obecně se tedy v této části budeme zabývat metodami, ve kterých při zpracování obrazu dochází k tomu, že jako vstup máme nějaký konkrétní obraz, ale výstup jsou pouze jeho části nebo atributy, které jsme získali z tohoto obrazu a které jsou pro naši následnou analýzu podstatné. [10]

### 1.1.1 Základní morfologické operace

Nejdříve se budeme věnovat v morfologii těmto dvěma pojmům: eroze a dilatace. Tyto operace jsou základem morfologického zpracování. Většina morfologických algoritmů, kterými se budeme zabývat, jsou založeny právě na těchto dvou operacích. [10] Jejich kombinováním můžeme dostat další operátory a jejich operace je vždy založená na použití tzv. strukturního elementu, což je maticová maska, která je většinou mnohem menší než zpracováváný obraz. Tato maska se posouvá po obraze jako lokální operátor. V tomto případě můžeme morfologické operátory zařadit mezi lokální nelineární operátory. [12]

Z jejich nelinearity vyplývá, že na ně nelze použít charakteristiky jako je například impulsní odezva. Na druhou stranu můžeme definovat jiné vlastnosti. Je užitečné analyzovat operátory těmito způsoby a využít tyto vlastnosti, když určujeme postup obrazového zpracování. Mezi nejdůležitější vlastnosti patří: invariance s prahovou dekompozicí, invariance s translací, idempotence, dualita a další. Masky morfologického operátoru je většinou binární a nejčastěji je maska velikosti 3x3, ale větší masky například nepravidelného tvaru také lze využít. [12]



Obr. 1.1: Dvě isotropické binární masky. Vlevo 4-spojené sousední pixely. Vpravo 8-spojené sousední pixely. Zdroj: [12]

Co se týče obsahu masky, na Obr. 1.1 můžeme vidět dvě isotropické binární masky, které jsou vytvořeny sousedními pixely. Tyto masky jsou používány nejčastěji, ale větší čtverce, kříže, řádky i nepravidelné tvary masky jsou využívány také. V každé masce značí počátek kříž. [12]

Každý morfologický operátor představíme v jeho binární formě. Nadále budeme uvádět výsledný obraz  $Y$  v novém prostoru stejných rozměrů jako vstupní obraz  $X$ , aby bylo vysvětlení dostatečně přehledné. [12]

## Eroze

Binární eroze pro každou prostorovou pozici  $(i, k)$  poskytuje binární výsledek. Když aplikujeme operátor eroze použitím masky  $H$  na vstupní obraz  $X$  v pozici  $(i, k)$ , kterou označíme  $x$ , potom maska  $H$  a posunutí masky, které je ale stále v oblasti počátku  $x$  můžeme označit  $H_x$ . Výsledný obraz poté bude dle rovnice: [12]

$$Y = E_H(X) = x | H_x \subseteq X \quad (1.1)$$

Binární eroze, jak už název napovídá, eroduje vstupní množinu, tedy několik vstupních pixelů. Tyto pixely jsou nastaveny na hodnotu nula, tedy zmenšují velikost množiny. Tato změna je nevratná, protože malé objekty a výstupky, které se nemohou přizpůsobit strukturnímu elementu byly navždy úplně odstraněny. Z toho plyne, že eroze také může způsobit potlačení šumu za cenu zmenšení obvodu objektu. [12]

## Dilatace

Binární dilatace poskytuje pro každou prostorovou pozici  $x = (i, k)$  binární výsledek. Pokud použijeme podobný zápis jako v případě eroze, můžeme výsledný obraz definovat jako: [12]

$$Y = D_H(X) = x | H_x \cap X \neq \emptyset \quad (1.2)$$

Dilatace zvyšuje hodnotu vstupní množiny, tedy několik vstupních pixelů o hodnotě nula na okrajích objektu. Tím se většinou myslí malé proužky kolem obvodu objektu. Tyto pixely jsou nastaveny na hodnotu jedna, čímž se zvýší velikost množiny. Z důvodu nelinearity je i tato změna, stejně jako v případě eroze, nevratná. Je nevratná, protože malé díry a otvory, které nemohou zahrnout strukturní element, aniž by se jeden nebo více jejich elementů neprotínalo s množinou obrazu, byly změněny pro nastavení pixelů. Z tohoto důvodu může být dilatace použita pro potlačení šumu pozadí, plnění malých děr a také může způsobit připojení odpojených objektů za cenu toho, že se zvětší obvod objektu. [12]

Z operátorů popsaných výše plyne, že eroze odstraňuje šum vně objektu, malé objekty a výstupky, zatímco dilatace se používá k tomu, abychom vyfiltrovali šum pozadí, malé díry a mezery. Nežádoucí vedlejší efekt obou těchto operátorů je změna rozměrů objektů, jak jsme popsali výše. [12]

## Morfologické otevření

Morfologické otevření značené  $O$  je operátor, který je tvořen kaskádou eroze a dilatace, což vyplývá z následující rovnice: [12]

$$Y = O_H(X) = D_H(E_H(X)) \quad (1.3)$$

Protože většina masek je symetrických, je použití strukturního elementu v druhé operaci velmi důležité. Morfologické otevření je poté nezávislé na pozici počátku masky. Otevření způsobí, že aktivní pixely mohou buď zůstat nebo být odstraněny a nic jiného. Důležité je stejně jako v případě eroze, jestli maska pasuje k množině v  $X$ . V tomto případě je ovšem odpověď na tento problém poněkud jiná. Zatímco u eroze pracujeme pouze s pixely, které jsou součástí počátku masky, v tomto případě musí být celá množina pokryta maskou naplněná hodnotami jedna. [12]

## Morfologické uzavření

Morfologické uzavření označované  $C$  je operátor vytvořený kaskádou dilatace a eroze. [12]

$$Y = C_H(X) = E_H(D_H(X)) \quad (1.4)$$

Použití strukturního elementu v druhé operaci je i v tomto případě velmi důležité, a to ze stejných důvodů jako u morfologického otevření. Morfologické zavření je extenzivní operace. Aktivní pixely zůstávají a několik dalších přibude. Zavření je také nezávislé na pozici počátku masky. V tomto případě se můžeme ptát, jestli pasuje maska k množině pozadí v  $X$ . Tentokrát je odpovědí nastavit všechny pixely pod maskou na hodnotu 0. [12]

Na závěr můžeme shrnout, že zatímco morfologické otevření odstraní všechny objekty nebo jejich části, které se nemohou přizpůsobit velikosti masky, morfologické zavření přiřadí hodnotu jedna všem strukturám v pozadí, které se mohou přizpůsobit velikosti masky, tedy přidá pixely pozadí k pixelům objektu. [12]

Mezi další morfologické operátory patří například: [12]

- Fit-and-miss operátor
- Odvozené operátory
- Geodezické operátory

### 1.1.2 Skelet

Dilatace a eroze jsou základními operátory v matematické morfologii. Strukturní element je předdefinovaný tvar, který se používá v morfologických operacích s obrazy, jak jsme již zmínili. Binární digitální skelet je také jedním z hlavních operátorů a může být vypočítán s pomocí základních morfologických operátorů. Je dokázáno, že podmnožiny skeletu  $S(X, nB)$   $n$ -tého řádu binárního obrazu mohou být spočítány s pomocí následujícího vzorce: [24]

$$S(X, nB) = X \ominus nB - (X \ominus nB) \circ B \quad (1.5)$$

Kde  $B$  je strukturní element. Ze skupiny podmnožin  $S(X, nB)$  a známého řádu  $n$  můžeme získat tvar původního binárního obrazu  $X$  dalšími matematickými operacemi. [24]

### 1.1.3 Kontura

Problém detekce kontur a segmentace je podobný, ale ne identický. V podstatě detektory kontur nezajišťují, že dostaneme uzavřenou oblast kontury, a tedy takové



detektory nemusí vždy zajišťovat rozdělení obrazu na regiony. Ovšem vždy se můžeme pokusit získat uzavřenou konturu z regionů ve formě jejich obrysů. Nyní se podíváme na to, jak je možné obrátit proces a získat regiony z detektorů kontur. Na to existuje několik postupů. [2]

Původním přístupem k detekci kontur bylo se zaměřit na měření přítomnosti obrysu v určité části obrazu pomocí lokálních měření. Například operátory, které detekují hrany tím, že konvolují obraz s lokálními derivačními filtry. Nebo použití tzv. nulových bodů laplaceovského gaussovského operátoru. Cannyho detektor [2] také modeluje hrany jako ostré diskontinuity jasů, čímž se přidávají nemaximální potlačení a kroky prahování hystereze. Lepší popis můžeme získat uvažováním odezvy obrazu na skupinu filtrů různých parametrů a orientací. Příkladem takového postupu je tzv. Energeticky orientovaný přístup, který využívá kvadrurní páry sudých a lichých symetrických filtrů. [2]

Nejvíce současné přístupy samozřejmě berou v potaz také barvu a informaci textury a používají se různé učící techniky. Martin et al. [2] definoval gradientní operátory pro kanály jasů, barvy a textur, a využil je jako vstup do klasifikátoru logistické regrese, aby získal informaci o síle hrany. Dollar et al. [2] navrhl tzv. BEL algoritmus (*Boosted Edge Learning algorithm*), který funguje na tom principu, že učí klasifikátor hran ve formě pravděpodobnostního stromu z tisíců jednoduchých funkcí vypočítaných z obrazu. Výhodou tohoto přístupu je, že je poté možné zvládnout například paralelismus a dokončení v počáteční fázi klasifikace. Maira et al. [2] vyrobil jednak obecné a jednak specifické detektory hran učením diskriminačních reprezentací lokálních částí obrazu. Pro každou třídu se algoritmus naučí diskriminační slovník a použije získanou rekonstruovanou chybu s každým slovníkem jako vstup funkce do konečného klasifikátoru. [2]

Rozeznáváme i mnoho dalších morfologických operací obrazu a proto jsou pro úplnost některé uvedeny zde: [10] [22]

- Hit-or-Miss transformace
- Vyplnění mezer
- Extrakce spojených komponent
- Konvexní obal
- Top-hat transformace
- Metoda tzv. prořezávání
- Bottom-hat transformace
- Morfologická rekonstrukce

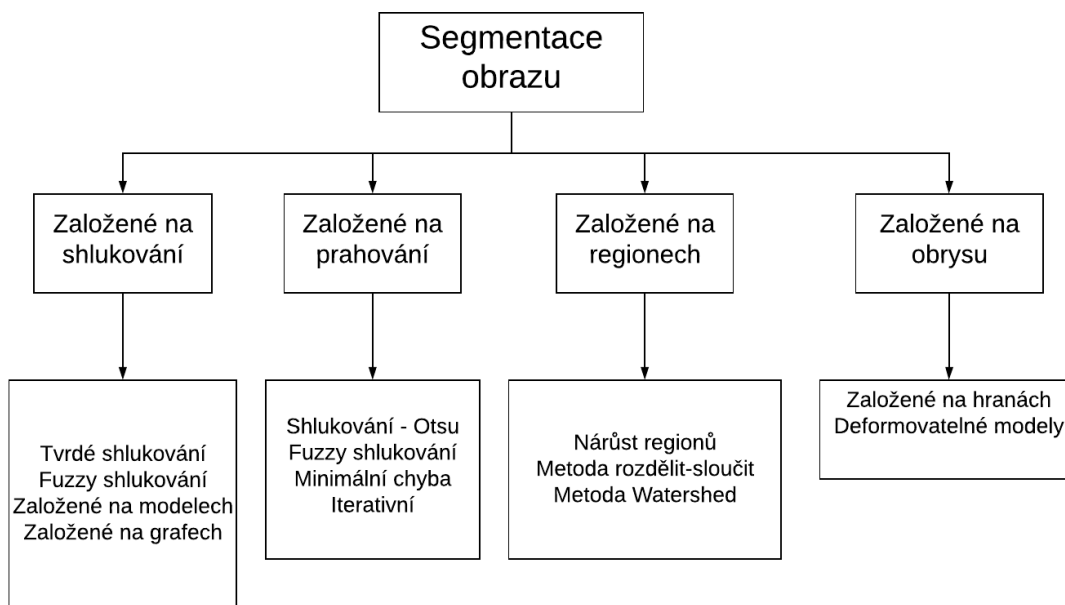
## 1.2 Segmentace obrazu

Pro metodu tvarové analýzy je klíčové rozdělit obraz na podstatné složky pro následný popis hledaných objektů. K tomu nám slouží segmentace obrazu. Segmentace obrazu je důležitým krokem v soustavě metod digitálního zpracování obrazu. Segmentace dělí obraz na jeho jednotlivé složky, popřípadě objekty. Kvalita dělení obrazu záleží na problému, který chceme metodou řešit. To v našem případě znamená, že segmentace obrazu by měla přestat ve chvíli, kdy v obrazu nalezneme objekty nebo regiony, které nás zajímají. Nemá tedy smysl provádět segmentaci obrazu do většího detailu, než je nutné pro správnou detekci elementů, které jsou pro nás důležité. [10]

Segmentace složitějších obrazů je jednou z nejnáročnějších oblastí zpracování obrazu, protože velmi záleží na její přesnosti, která následně určuje její úspěch či neúspěch. Z tohoto důvodu je obzvlášť důležité dbát zvýšené pozornosti, abychom dosáhli co nejlepší pravděpodobnosti správné segmentace. V praxi je důležité pro daný obraz vybrat pouze ty analyzátory obrazu, které vyznačí oblasti zájmu a zároveň takové, které potlačí zbytečné detaily obrazu. [10]

Většina známých algoritmů pro segmentaci jsou založené na dvou základních vlastnostech: diskontinuita a podobnost. V prvním případě nám jde o to, abychom rozdělili obraz na základě prudkých změn jeho intenzity. Mezi tento princip patří například detekce hran obrazu. V druhém případě je principem rozdělení obrazu na regiony, které jsou podobné z hlediska přednastavených kritérií. Do této kategorie můžeme zařadit metody jako prahování, růst regionů, dělení či průnik regionů. V segmentaci můžeme dosáhnout výrazných zlepšení, pokud dané metody vhodně zkombinujeme. Klasickým příkladem je kombinace metody prahování s detekcí hran. [10]

Segmentace obrazu ovšem není hlavní náplní této práce, a proto je níže uvedeno stručně pouze základní rozdělení těchto metod na Obr. 1.2 a následně popíšeme několik příkladů.



Obr. 1.2: Přehled metod segmentace obrazu. Zdroj: [7]

### 1.2.1 Detekce hran

Detekce hran je segmentace, při které hledáme pixely obrysu regionu. Nalézáme hrany procházením sousedních pixelů. Obrys regionu tvoří tyto hrany a my měříme rozdílné hodnoty mezi sousedními pixely. Detekci hran můžeme formulovat jako binární klasifikaci na úrovni pixelů s cílem identifikovat jednotlivé pixely buď jako tzv. *on-edge*, tedy součástí hrany, nebo *off-edge*, tedy mimo hranu. S tímto případem, kdy se jedná o binární rozdělení, budeme dále pracovat. [13]

Hrana může být popsána jako hranice mezi sousedními oblastmi obrazu. Detekce hran je velmi užitečný a často používaný nástroj ve zpracování obrazu. Odlišné aplikace mají jiné nároky na detekci hran, které jsou na jednu stranu většinou informační, tedy nám stačí pouze několik širokých záběrů pro to, abychom zachytili hlavní podstatu obrazu. Na druhou stranu abychom v obraze zachytili každý malý detail, je důležité vyžadovat jiný přístup k problematice. Z tohoto důvodu neexistuje úplná definice a vzniklo spoustu různých algoritmů k detekci hran. Také je obtížné kvantifikovat výkonost těchto algoritmů smysluplně. [13]

### 1.2.2 Segmentace založená na regionech

V této metodě segmentace pixely korespondující objektu tvoří skupiny a jsou označeny. Důležitým principem je hodnota podobnosti, která vypovídá o rozdílech v hodnotách pixelů, a také odchylka těchto hodnot. Dalším principem je pak prosto-

rová vzdálenost, která se skládá z euklidovské vzdálenosti a kompaktnosti regionu. Pro správnost metody této segmentace je také potřeba použít vhodnou metodu prahování. Předpokládáme, že body na konkrétním objektu se zobrazí na prostorově blízkých pixelech obrazu, které mají obdobné hodnoty. [13]

### 1.2.3 Obrysové trasování

Tato metoda segmentace obrazu je založená na detekci ostrých, lokálních změn v intenzitě. [10] Je to přirozený postup, který aplikujeme na reprezentaci hrany tam, kde je obrys rozpojený nebo vícenásobný. Většinou ovšem potřebujeme dodatečné informace o obraze, aby bylo možné trasovat na základě podobnosti mezi pixely, které do části obrysu patří. Potom binární reprezentace hrany  $b$ , kde hodnota 1 odpovídá pixelům hrany a hodnota 0 odpovídá pixelům ostatního, by byla doplněna jednak intenzitou hrany  $e$  a informací o směru  $\phi$ . Tedy použití vektorové reprezentace hrany jako vstup  $((b, e, \phi) - \text{vektor na pixel})$ . Také musíme zabránit trasování úplně falešného obrysu, které začne od falešného bodu hrany. To může nastat při označení takového bodu z důvodu šumu. Potřebujeme nějakou primární informaci o obrysech nebo objektech. Například v takové formě, že zahájíme trasování interaktivně nebo na základě předešlých hrubých segmentací za použití jiných metod. [12]

Ještě předtím než popíšeme algoritmus, bychom měli definovat použitý pojem sousedství a také míru podobnosti. Sousedství může být buď spojené po čtyřech nebo po osmi a je považováno pro popis připojení sousedních pixelů. Míra podobnosti mezi  $i$ -tým počátečním pixelem a  $(i + 1)$ -tým pixelem může být definována kvantitativně délkou  $L$  vektoru rozdílnosti. [12]

$$L = \sqrt{(a\Delta e)^2 + (\Delta\varphi)^2}, \quad \Delta e = |e_i - e_{i+1}|, \quad \Delta\varphi = |\varphi_i - \varphi_{i+1}| \bmod 2\pi \quad (1.6)$$

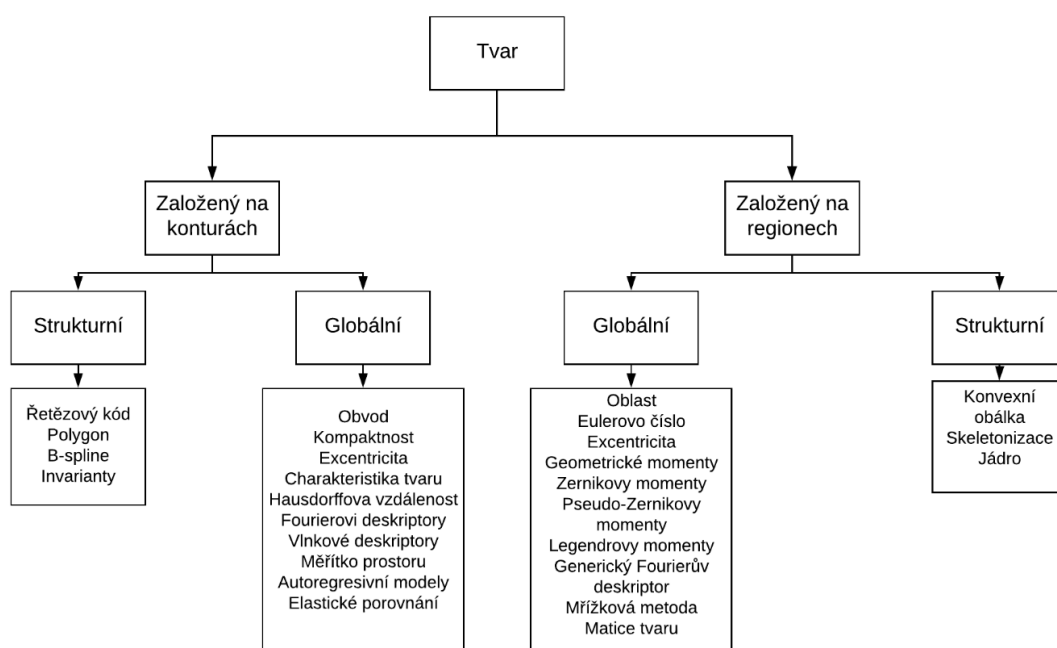
V tomto vzorci zvolený koeficient  $a$  balancuje vliv obou rozdílností. [12]

Když je počáteční pixel obrysu vybrán, následný princip trasování je jednoduchý. U označených pixelů  $b$  ve zvolené sousední oblasti kolem již potvrzeného bodu je poté měřena míra podobnosti s tímto počátečním pixelem. Nejvíce podobný sousední pixel je poté testován pro absolutní podobnost a pokud je výsledná hodnota vyšší než daný práh, nový pixel je přijat jako pokračování obrysu. Následně je algoritmus opakován s novým pixelem jako startovní bod dokud celá cesta trasování neskončí, tedy už nemá pokračování. To nastane pokud nejsou žádné označené sousední pixely nebo pokud žádný pixel nesplňuje minimální hodnotu podobnosti. Obrys se tedy uzavře nebo trasa dosáhne k okraji obrazu. Může být vybráno několik počátečních startovních bodů zároveň, abychom našli obrysy několika objektů nebo abychom našli několik částí obrysů. [12]

## 1.3 Reprezentace a popis obrazu

Poté co byl obraz segmentován na několik regionů metodami, které jsme popsali, je výsledný souhrn segmentovaných pixelů popsán formou, která je vhodná pro následné zpracování počítačem. Kolem této problematiky můžeme učinit dvě rozhodnutí. Buď region v obraze reprezentujeme na základě jeho vnějších charakteristik jako je například jeho obrys, nebo ho můžeme reprezentovat na základě jeho vnitřních charakteristik, to znamená pixelů, které jsou v regionu zahrnuty. Výběr jedné z těchto možností je ovšem pouze základem při celkovém zpracování dat počítačem. Dalším cílem je popsat tyto regiony na základě zvolené reprezentace. Názorným příkladem je, že region může být reprezentován jeho obrysem a tento obrys můžeme popsat dále například pomocí jeho délky, počet konkávních bodů v obrysu a s pomocí dalších. Funkce zvané deskriptory, tedy které popisují region, musí být v některých případech dostatečně necitlivé vůči různým variacím rozměrů, rotacím a dalším faktorům podle potřeby. [10]

Techniky reprezentace tvaru můžeme rozdělit podle následujícího diagramu na Obr. 1.3.



Obr. 1.3: Klasifikace reprezentace a popisu tvaru. Zdroj: [28]

V našem případě, kdy se zabýváme tvarovou analýzou, nás bude zajímat především reprezentace pomocí vnějších charakteristik, a proto se dále budeme zabývat více touto reprezentací, tedy zejména reprezentací na základě kontur. Nicméně je

dobré se zmínit i o metodách založených na regionech a právě proto jsou zobrazeny na Obr. 1.3.

### 1.3.1 Reprezentace tvaru na základě kontur

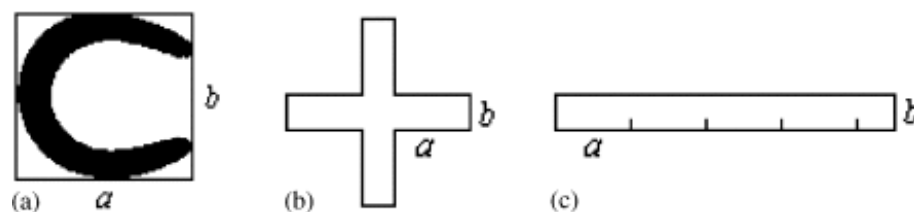
Tvarové klasifikace na základě kontur využívají pouze informace obrysu tvaru. Obecně rozlišujeme dva velmi odlišné postupy modelování tvaru: kontinuální přístup (globální) a diskrétní přístup (strukturní). Kontinuální přístupy nerozdělují tvar na dílčí části. Většinou používáme pro popis tvaru vektor příznaků derivovaný z integrální hranice. Míra podobnosti tvaru je většinou metrika vzdálenosti mezi získanými vektory vlastností. Diskrétní přístupy vytvoří z obrysu tvaru segmenty, které se nazývají primitivy, s pomocí konkrétního kritéria. Konečná reprezentace je většinou řetězec nebo graf. Míra podobnosti je poté určena porovnáním řetězců nebo porovnáním grafů. [28]

### 1.3.2 Globální metody

Reprezentace tvaru na základě globálních kontur většinou počítají vícerozměrný číselný vektor příznaků z informace tvaru obrysu. Porovnání tvarů je přímočarý postup, který se obvykle skládá z použití metriky vzdálenosti jako je euklidovská vzdálenost. Porovnání prvků (prvek příznaku) se také používá v některých aplikacích. [28]

#### Nejjednodušší tvarové deskriptory

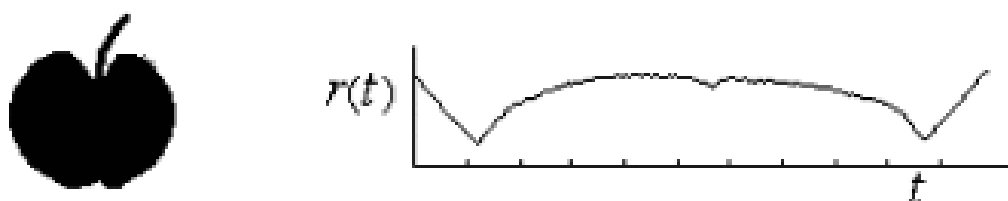
Běžné jednoduché globální deskriptory jsou obsah, kruhovitost, excentricita. Kruhovitost můžeme vyjádřit jako  $obvod^2/obsah$ . Excentricitu poté jako  $délku\ hlavní\ osy/délku\ vedlejší\ osy$ . Dále mezi tyto deskriptory můžeme zařadit například orientaci hlavní osy a tzv. ohybovou energii (*bending energy*). Tyto jednoduché globální deskriptory mohou obvykle rozlišit pouze tvary s velkým rozdílem, tedy se většinou používají jako filtry k eliminaci falešných detekcí nebo se kombinují s jinými tvarovými deskriptory pro rozlišení tvarů. Nejsou tedy přizpůsobeny k tomu být jedinými tvarovými deskriptory. Například excentricita tvaru na Obr. 1.4 (a) je blízko hodnotě 1. To ovšem nepopisuje tvar správně, protože jak vidíme, jedná se o protáhlý tvar. V tomto případě je kruhovitost lepší deskriptor. Další dva tvary na Obr. 1.4 (b) a (c) mají stejnou kruhovitost, ale jedná se o dva různé tvary. V tomto případě je lepším deskriptorem excentricita. Peura a Iivarinen [28] navrhli i další jednoduché deskriptory založené na globálních konturách. Mezi ně patří konvexita, poměr hlavní osy, kruhová a eliptická variance. [28]



Obr. 1.4: Excentricita a kruhovitost tvaru. Zdroj: [28]

### Tvarová charakteristika

Tvarová charakteristika představuje tvar pomocí jednorozměrné funkce odvozené z hraničních bodů tvaru. Existuje mnoho tvarových charakteristik, které zahrnují centroidální profil, komplexní souřadnice, centroidní vzdálenost, kterou můžeme pozorovat na Obr. 1.5. Mezi další tvarové charakteristiky patří tečný úhel, kumulativní úhel, zakřivení, plocha a další. Tvarové charakteristiky jsou obvykle normalizovány, aby byly invariantní vůči translaci a škálování. Aby bylo možné kompenzovat změny orientace, je třeba posouvat porovnávání, abychom našli nejlepší shodu mezi dvěma tvary. Většina porovnávacích charakteristik je normalizována pro posouvání porovnávání v 1-D prostoru, nicméně některé vyžadují porovnávání ve 2-D prostoru, jako je porovnání centroidálních profilů. [28]



Obr. 1.5: Tvar jablka a jeho centroidní vzdálenost. Zdroj: [28]

Kromě vysokých nároků porovnávání jsou tvarové charakteristiky citlivé na šum a i malé změny v obrysu mohou způsobit velké chyby při porovnávání. Proto je nežádoucí pomocí tvarové charakteristiky přímo popisovat tvar. Další zpracování je nutné ke zvýšení robustnosti a snížení zátěže ohledně porovnávání. Například může být tvarová charakteristika zjednodušena kvantizací charakteristiky do histogramu charakteristiky, který je invariantní vůči rotaci. [28]

### Fourierovy deskriptory

V této části si popíšeme jednodimenzionální fourierův deskriptor. Tento deskriptor se velmi hojně využívá v aplikacích reprezentace tvaru zvláště k rozpoznání povahy daného tvaru. Jednoduché charakteristiky fourierova deskriptoru (dále jen FD) jako je

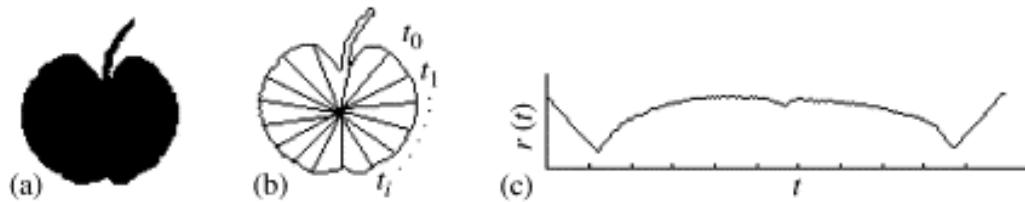
jednoduchá derivace, jednoduchá normalizace, odolnost vůči ruchu, vjemová smyslnost, hierarchie hrubého až jemného zobrazení a další, činí tento deskriptor tvaru velmi populární. Obecně získáme jednodimensionální (dále jen 1-D) FD po aplikaci fourierovi transformace (dále jen FT) na tvarovou funkci, která je derivována ze souřadnic  $(x(t), y(t))$ ,  $t = 0, 1, \dots, N - 1$  obrysu tvaru. Typická tvarová funkce je distanční funkce centroidu, která je dána vzdáleností bodů obrysu od centroidu  $(x_c, y_c)$  daného tvaru. [27]

$$r(t) = ([x(t) - x_c]^2 + [y(t) - y_c]^2)^{1/2}, t = 0, 1, \dots, N - 1, \quad (1.7)$$

kde  $x_c$  a  $y_c$  jsou dané vzorci: [27]

$$x_c = 1/N \sum_{t=0}^{N-1} x(t), \quad y_c = 1/N \sum_{t=0}^{N-1} y(t) \quad (1.8)$$

Na Obr. 1.6 je příklad distanční funkce centroidu tvaru jablka. [27]



Obr. 1.6: (a) Tvar jablka; (b) obrys jablka (a); (c) distanční funkce centroidu jablka (a). Zdroj: [27]

Jednodimensionální FT je tedy aplikováno na funkci  $r(t)$  pro získání fourierově transformovaných koeficientů. [27]

$$a_n = 1/N \sum_{t=0}^{N-1} r(t) \exp - j2\pi n t / N, n = 0, 1, \dots, N - 1 \quad (1.9)$$

Velikosti koeficientů  $a_n$  ( $n = 0, 1, \dots, N - 1$ ) normalizovaných velikostí prvního koeficientu  $a_0$  jsou použity jako tvarové deskriptory, které se nazývají fourierovy deskriptory. Tyto získané deskriptory jsou invariantní vůči translaci, rotaci i škálování. Je dokázáno, že reprezentace tvaru za použití fourierových deskriptorů je výkonnější než mnoho jiných tvarových deskriptorů. Ovšem všechny tyto metody předpokládají informaci o obrysu tvaru, která nemusí být dostupná ve všech situacích. Například je obtížné derivovat 1-D FD pro tvar na Obr. 1.7 (a), protože obrys tohoto tvaru není dostupný. Dále 1-D FD nemůže zachytit tvar vnitřního obsahu, což je důležité pro tvarovou klasifikaci. Jako další příklad uvedeme další dva obrázky na Obr. 1.7. FD není schopno rozlišit tvar na obrázku (b) od tvaru na obrázku (c). Tyto nevýhody limitují aplikaci 1-D FD. [27]





Obr. 1.7: (a) Tvar bez obrysu; (b), (c) dva tvary se stejným obrysem, ale jiným vnitřním obsahem. Zdroj: [27]

## Deskriptory založené na vlnkové transformaci

Podobně jako FT používá WT (*Wavelet transformation* - vlnková transformace) k popisu daného signálu elementární funkce, které se nazývají vlnky. Na rozdíl od FT, který používá harmonické funkce s různými frekvencemi, používá WT k odvození rekonstrukčních signálů pouze jednu základní vlnku (tzv. mateřskou vlnku). Dilatací, kompresí a posunem mateřské vlnky můžeme poté odvodit nové varianty tohoto signálu, které společně tvoří tzv. vlnkovou stavebnici (*wavelet building set*). Obecný derivační vzorec vlnek  $\psi^{a,b}(t)$  z mateřské vlnky  $\psi(t)$  je uveden takto: [17]

$$\psi^{a,b}(t) = |a|^{-1/2} \psi\left(\frac{t-b}{a}\right), \quad (1.10)$$

kde  $a$  je parametr komprese nebo dilatace a  $b$  je parametr posunutí. [17]

Na základě rovnice (1.10) ukazuje rovnice (1.11) koeficienty kontinuální vlnkové transformace  $W_\psi f^*(a, b)$  pro odvozenou úhlovou funkci zadanou jako  $f^*(t)$ . Tyto koeficienty budeme nazývat vlnkové deskriptory (WD z *wavelet descriptors*) podobné názvu fourierových deskriptorů (FD). Rozlišujeme například Haarův vlnkový deskriptor a Mexican Hat vlnkový deskriptor. Na základě MRA (*multiresolution analysis* - multiresoluční analýza) dostáváme přibližný signál pro  $a > 1$  a podrobný signál pro  $a < 1$ . Protože jsou vlnky časově omezenými variantami základní funkce, můžeme omezit integraci v rovnici (1.11) na definiční interval  $[0, 2\pi]$  a nakonec obdržet následující: [17]

$$W_\psi f^*(a, b) = |a|^{-1/2} \int_0^{2\pi} f^*(t) \psi\left(\frac{t-b}{a}\right) dt \quad (1.11)$$

## Statistické momenty

Tvar obrysu segmentů můžeme popsat kvantitativně s pomocí tzv. statistických momentů jako jsou průměr, variance nebo momenty vyššího řádu. Abychom si více přiblížili tuto metodu, považujme následující obraz jako jednodimenzionální (dále jen 1-D) funkci  $g$ . [10]

Tuto funkci získáme spojením dvou krajních bodů segmentu linií bodů a poté následnou rotací, dokud krajní body nejsou horizontálně, což můžeme pozorovat vpravo na obrázku. Souřadnice ostatních bodů jsou potočeny o stejný úhel. Amplitudu funkce  $g$  si můžeme představit náhodnou diskretní proměnou  $v$ , a díky tomu sestojit histogram amplitud  $p(v_i), i = 0, 1, 2, \dots, A - 1$ , kde  $A$  je množství diskretních amplitudových přírůstků. Důležité je podotknout, že histogram  $p(v_i)$  je pouze odhad pravděpodobnosti hodnoty  $v_i$ . [10]

Z těchto podmínek můžeme vyvodit vzorec, že: [10]

$$\mu_n(v) = \sum_{i=0}^{A-1} (v_i - m)^n p(v_i) \quad (1.12)$$

kde

$$m = \sum_{i=0}^{A-1} v_i p(v_i) \quad (1.13)$$

Kvantita  $m$  je určena průměrem hodnoty  $v$  a  $\mu_2$  jako jeho odchylkou. V podstatě nám stačí pouze prvních pár momentů, abychom od sebe rozlišili nápadně jasné tvary. [10]

Alternativní metodou je normalizovat funkci  $g(r)$ . V tomto případě vnímáme  $g(r_i)$  jako pravděpodobnost hodnoty  $r_i$ . Proměnou  $r$  vnímáme jako náhodnou a momenty jsou v tomto případě: [10]

$$\mu_n(r) = \sum_{i=0}^{K-1} (r_i - m)^n g(r_i) \quad (1.14)$$

kde

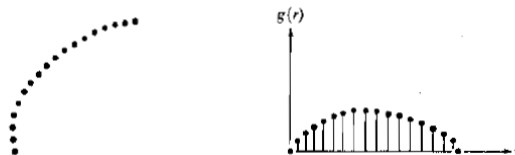
$$m = \sum_{i=0}^{K-1} r_i g(r_i) \quad (1.15)$$

V tomto vzorci proměnná  $K$  zastupuje počet bodů na obrysu a  $\mu_n(r)$  přímo souvisí s tvarem funkce  $g(r)$ . V tomto případě druhý moment  $\mu_2(r)$  udává rozpětí křivky přibližně podle průměru hodnoty  $r$ . Třetí moment  $\mu_3(r)$  udává symetrii křivky také na základě průměru. [10]

V podstatě jsme dosáhli toho, že jsme zúžili otázku popisu na 1-D funkce. Přestože momenty jsou zdaleka nejpoužívanější metodou, nejsou jedinými deskriptory používanými pro tento účel. [10]

Na závěr je důležité upozornit, že metoda momentů má výhodu oproti ostatním metodám v tom, že provedení této metody je více přímočaré a také v sobě nesou informaci o tvaru obrysu ve více uchopitelné formě. Také další výhodou je, že tato

metoda je velmi nezávislá vůči faktorům, o kterých jsme se už zmiňovali výše jako je například rotace. Normalizace rozměrů lze dosáhnout upravováním hodnot  $g$  a  $r$ . [10]



Obr. 1.8: Segment obrysu a reprezentace jako 1-D funkce. Zdroj: [10]

Je celá řada dalších deskriptorů, kterými se nebudeme zabývat, ale pro úplnost jsou zde vypsány další známé příklady: [28]

- Tvarové deskriptory založené na tzv. korespondenci
- Obrysové momenty
- Pružné porovnávání
- Stochastická metoda
- Metoda škálování

### 1.3.3 Strukturní metody

Další součástí tvarové analýzy jsou strukturní reprezentace tvaru. Jak se již psalo výše, při tomto přístupu jsou tvary rozděleny na segmenty obrysu, které se nazývají primitivy. Různé strukturní metody se liší ve výběru primitiv a v organizaci primitiv pro reprezentaci tvaru. Běžné metody dekompozice obrysu jsou založeny na polygonální aproximaci, dekompozici křivky a jejím přizpůsobení. Výsledek je zakódován do řetězce obecné formy: [28]

$$S = s_1, s_2, \dots, s_n, \quad (1.16)$$

kde  $s_i$  může být element řetězového kódu, strany polygonu, tzv. *spline* atd.  $s_i$  může obsahovat několik atributů jako délka, průměrné zakřivení, maximální zakřivení, ohybovou energii, orientaci atd. Řetězec může být použit přímo pro popis nebo může být použit jako vstup do syntaktického analyzátoru vyšší úrovně. Nyní se podíváme na metody tvarové reprezentace a popisu s použitím  $S$ . [28]

#### Reprezentace řetězovým kódem

Řetězový kód popisuje objekt sekvencí čísel, které zastupují segmenty jednotkové délky s danou orientací. Metoda byla představena v roce 1961 Freemanem [28],

který popsal metodu umožňující kódování libovolných geometrických konfigurací. Libovolná křivka je reprezentována sekvencí malých vektorů jednotkové délky a omezeným souborem možných směrů a proto je tato metoda nazývána metoda jednotkového vektoru. Při implementaci je digitální obrys obrazu překrýván mřížkou. Body obrysu jsou aproximovány k nejbližšímu bodu mřížky a získáme vzorkovaný obraz. Z vybraného počátečního bodu lze řetězový kód vygenerovat pomocí 4-směrného nebo 8-směrného řetězového kódu. N-směrný (tedy kde  $N > 8$ ) řetězový kód je také možný a nazývá se obecný řetězový kód. [28]

Pokud je řetězový kód použit pro porovnávání, musí být nezávislý na výběru prvního pixelu obrysu v sekvenci. Jednou z možností normalizace řetězového kódu je najít pixel v hraniční sekvenci, z čehož jako výsledek vyplyne minimální celé číslo, pokud je řetězec popisu interpretován jako základní čtyř číselný. Tento pixel je pak použit jako počáteční pixel. Alternativně může být obrys reprezentován rozdíly v po sobě jdoucích směrech v řetězovém kódu namísto reprezentování obrysu relativními směry. Toto lze vypočítat odečtením každého prvku řetězového kódu od předchozího a převzetím výsledného modulo  $n$ , kde  $n$  je konektivita. Po těchto operacích je rotačně invariantní řetězový kód získán cyklickou permutací, která produkuje nejmenší číslo. Takový normalizovaný diferenciální řetězový kód se nazývá číslo tvaru (*shape number*). Takto odvozený řetězový kód není invariantní vůči škálování. Ačkoli je možné škálovat dva podobné tvary do stejné velikosti, výsledná čísla tvarů mohou mít různý počet číslic, takže je nepraktické provádět porovnání mezi dvěma takovými tvary. [28]

Řetězový kód má obvykle vysoké rozměry a je citlivý na šum. Často se používá jako vstup do vyšší úrovně analýzy. Například může být použit pro aproximaci polygonu a pro nalezení zakřivení obrysu, což je důležitá percepční vlastnost. [28]

Iivarinen a Visa [28] odvozují histogram řetězového kódu (CCH z *chain code histogram*) pro rozpoznávání objektů. CCH se počítá jako: [28]

$$p(k) = n_k/n \quad (1.17)$$

kde  $n_k$  je počet hodnot kódu řetězce  $k$  v řetězovém kódu a  $n$  je počet článků v řetězovém kódu. CCH odkazuje na pravděpodobnost různých směrů přítomných v obrysu. CHH je invariantní vůči translaci a škálování, ale je invariantní pouze vůči rotaci o 90 °. Proto se navrhuje normalizovaný CHH (NCHH). Je definován jako: [28]

$$p(k) = l_k n_k / l \quad (1.18)$$

kde  $n_k$  je to stejné jako u CHH,  $l_k$  je délka směru  $k$  a  $l$  je délka kontury. Přestože CHH zmenšuje rozměry reprezentace řetězového kódu, neřeší problém citli-

vosti na šum. [28]

## Rozklad polygonu

Obrys tvaru je rozdělen na segmenty polygonovou aproximací. Vrcholy polygonu se používají jako primitiva. Prvek pro každý primitiv je vyjádřen jako čtyřprvkový řetězec, který se skládá z vnitřního úhlu, vzdálenosti od následujícího vrcholu a jeho souřadnic  $x$  a  $y$ . Je zřejmé, že tato funkce není invariantní vůči translaci, škálování a rotaci. Podobnost mezi libovolnými dvěma tvary je vzdálenost dvou řetězců prvků. Z důvodu účinnosti a robustnosti je z každého tvaru vybrán pouze pevný počet nejostřejších vrcholů. Proto je pro index prvků generována kolekce prvků patřících všem modelům v databázi. Prvky jsou poté uspořádány do binárního stromu nebo do  $m$ -nárního stromu. Porovnání tvarů zahrnuje dva kroky. Přizpůsobení podle prvků v prvním kroku a přizpůsobení podle modelů ve druhém kroku. V prvním kroku, vzhledem k vlastnostem dat ve tvaru otazníku, je prvek prohledán indexovaným stromem. Pokud se zjistí, že určitá vlastnost modelu v databázi je podobná vlastnostem dat prvku, potom seznam tvarů přidružených k vlastnostem modelu je získán. Ve druhém kroku se shoda mezi tvarem otazníku a získaným modelem porovná na základě editační vzdálenosti mezi dvěma řetězci primitiv. [28]

Mehrotra a Gary [28] představovali tvar jako řetězec vektorů. Pro tvar je z polygonální aproximace obrysu tvaru detekována řada bodů zájmu. Vzhledem k tvaru s  $n$  body zájmu je vybrán pár tak, aby tvořil základní vektor. Základní vektor je normalizován jako jednotkový vektor podél osy  $x$ . Všechny ostatní body zájmu tvaru jsou transformovány do tohoto souřadnicového systému. Tvar je pak reprezentován sadou souřadnic  $(x_1, y_1), \dots, (x_n, y_n)$ , kde  $(x_i, y_i)$  jsou normalizované souřadnice  $i$ -tého bodu zájmu (Obr. 1.9). Do sady souřadnic se také přidají čtyři transformační parametry, aby se vytvořil konečný vektor vlastností, čtyři parametry jsou  $S$ ,  $T_x$ ,  $T_y$ ,  $\theta$ , které představují měřítko, translaci a úhel základního vektoru. Pro dosažení nezávislosti počátečního bodu jsou všechny vektory vytvořené spojením dvou sousedních bodů použity jako základní vektory postupně. Podobnost mezi dvěma vlastnostmi se měří euklidovskou vzdáleností. Vektory prvků obrysu jsou uspořádány do  $k$ -rozměrného  $B$ -stromu. Pro danou vlastnost tvaru otazníku je hledán index a je vytvořen seznam tvarů s podobnými vlastnostmi jako jsou vlastnosti otazníku. Porovnání jedné nebo více vlastností nezaručuje úplnou shodu tvaru. V důsledku toho, jakmile jsou získány tvary s podobnými vlastnostmi, je podobnost tvaru zkontrolována překrýváním každého získaného tvaru s tvarem otazníku a vyhodnocením míry překrývání mezi nimi.



Obr. 1.9: Řetězové vektory: (a) původní popis obrysu; (b) normalizovaný popis obrysu. Zdroj: [27]

Všechny výše uvedené tři metody aproximovaly tvar jako polygon a tvar je reprezentován jako řetězec segmentů, které jsou potom uspořádány do stromové datové struktury. I když se očekává, že toto bude dobře fungovat pro uměle vytvořené objekty, použití pro přírodní objekty je nepraktické. [27]

Stejně jako u globálních metod i v tomto případě existuje celá řada dalších deskriptorů, které je dobré zmínit: [27]

- Hladký rozklad křivky
- Metoda škálování
- Syntaktická analýza
- Tvarové invarianty

### 1.3.4 Geometrické deskriptory

V této části se podíváme blíže na popis a analýzu tvaru konkrétních geometrických obrazců, které se nám budou později hodit při řešení na konkrétních datech. Uvedeme pouze několik geometrických deskriptorů, abychom přiblížili princip měření v obrazových datech. Do těchto deskriptorů se samozřejmě počítají i základní geometrické parametry jako je obsah, obvod a další, které zde nebudeme pro základní geometrické tvary rozsáhle uvádět, ale uvedeme některé jiné zajímavé způsoby získání deskriptorů. Existuje také velké množství dalších deskriptorů jiných tvarů a jiných vlastností, které nebudeme uvádět.

#### Nejjednodušší deskriptory

Délka určitého obrysu regionu je jedním z nejjednodušších deskriptorů. Počet pixelů, které tvoří hranici, zhruba odpovídá délce tohoto obrysu. Potom můžeme říct, že tzv. *diameter* obrysu  $B$  je definován jako: [10]

$$Diam(B) = \max_{i,j} [D(p_i, p_j)] \quad (1.19)$$

kde  $D$  odpovídá naměřené vzdálenosti a  $p_i, p_j$  odpovídají bodům křivky, které udávají obrys regionu. [10]

Pokud se jedná o uzavřený objekt, hovoříme o obvodu, který vyjadřuje právě i délku celkového obrysu dvojrozměrného tvaru. Označuje se písmenem  $o$ . Například pro čtverec platí jednoduchý vzorec  $o = 4a$ . [23]

Obsah vyjadřuje velikost dvojrozměrného prostoru, který objekt zabírá. Většinou ho označujeme písmenem  $S$ . Existuje několik základních vzorců pro výpočet různých geometrických obrazců, které dále využijeme při praktické části řešení pro klasifikaci tvarů v obrazu. Například pro čtverec platí jednoduchý známý vzorec  $S = a^2$ . [23]

## Kruhovitost

Kruhovitost nebo kulatost je velmi hojně využívané měřítko v oblasti 2D obrazů a můžeme ji vyjádřit následujícím vzorcem: [1]

$$Kruhovitost = p^2/4\pi A \quad (1.20)$$

V tomto vzorci  $p$  odpovídá obvodu a  $A$  obsahu 2D celé částice. V případě kruhu by byl vzorec roven hodnotě 1, zatímco objekty jiných tvarů budou mít hodnotu větší než 1. [1]

## Elipticita

Jeden z možných přístupů získání tohoto deskriptoru je založený na momentových invariantách. Jelikož každou elipsu můžeme získat tak, že provedeme afinní transformaci kruhu, používáme nejjednodušší afinní momentovou invariantu kruhu, abychom charakterizovali elipsu. Tato invarianta je založená na centrálním momentu  $\mu_{pq}$ . [19]

$$I_1 = \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{00}^4} \quad (1.21)$$

Všechny kruhy pak budou vycházet se stejnou hodnotou  $I_1$ . Problém nastává v tom, že i jiné tvary než kruh mohou také vyjít se stejnou hodnotou. Abychom takovým problémům předešli, je nutné tvar kruhu více oddělit od ostatních. Toho dosáhneme tak, že do problému zavedeme invarianty vyššího řádu. Vyšší řády jsou ovšem méně spolehlivé. Například výše uvedený vzorec pro  $I_1$  pracuje pouze s momenty druhého řádu, a je tedy více praktický. [19]

Hodnotu invarianty  $I_1$  můžeme také vyjádřit jako: [19]

$$I_1 = \frac{1}{16\pi^2} \quad (1.22)$$

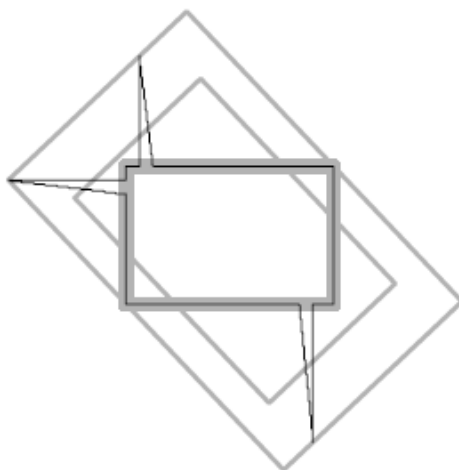
Následně můžeme vyjádřit míru elipticity jako: [19]

$$E_I = \begin{cases} 16\pi^2 I_1, & \text{if } I_1 \leq \frac{1}{16\pi^2} \\ \frac{1}{16\pi^2 I_1}, & \text{jinak} \end{cases} \quad (1.23)$$

Rozsah hodnot je potom mezi 0 a 1, kde hodnota 1 zastupuje dokonalou elipsu. Mezi další přístupy k popisu elipticity patří: eliptická odchylka, euklidovská elipticita a další. [19]

### Tvar obdélníku

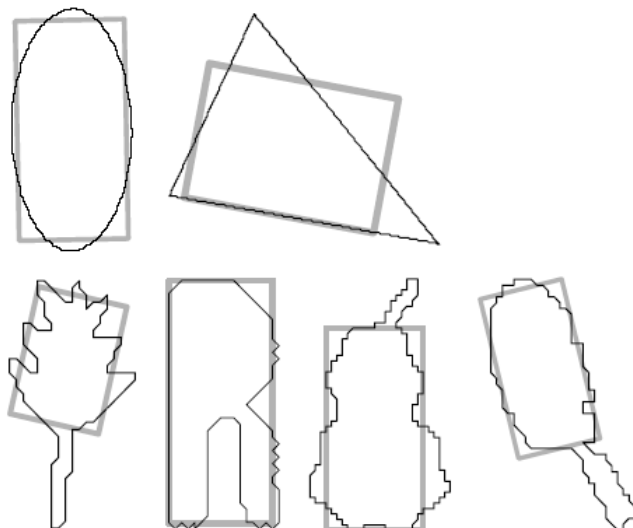
Základním přístupem k měření tvaru obdélníku je metoda nazývaná MBR (*minimum bounding rectangle* - minimum ohraničující obdélník). Název je odvozený od toho, že využíváme poměru oblasti regionu ku oblasti minima ohraničujícího obdélník. MBR konvexního mnohoúhelníku může být vypočten pomocí tzv. Toussaintovi metody. [19] Tato metoda vyjadřuje, že jelikož konvexní obal jednoduchého mnohoúhelníku můžeme nalézt v čase lineárně závislém, tak celý algoritmus metody je také lineární. [19]



Obr. 1.10: Standardní MBR a její škálovaná verze. Zdroj: [19]

Jednou z možností, jak předejít přílišné citlivosti metody MBR je tzv. robustní MBR. V tomto případě povolíme v požadavku MBR, že musí obsahovat všechny body. Pokud najdeme MBR, které potřebuje obsahovat pouze většinu regionu, ale ne všechny body, potom by tento MBR měl být silnější v přítomnosti malých odchylek obrysu. [19]





Obr. 1.11: Přilnutí robustního MBR k různým tvarům. Zdroj: [19]

Formulace kritéria je velmi podobná předešlé, ale mírně upravena: [19]

$$\frac{R + D}{I} \quad (1.24)$$

Jmenovatel  $I$  odpovídá oblasti průsečíku regionu a obdélníku. Díky tomu obdélník obsahuje co nejvíce dat, ale zároveň se snažíme, aby byl obdélník co nejmenší. Robustní MBR tedy získáme tak, že začneme se standardní MBR metodou, přičemž její rozměry škálujeme na polovinu původní velikosti. Poté iterativně zmenšujeme  $\frac{R+D}{I}$ . Nakonec můžeme vyjádřit míru podoby s obdélníkem jako: [19]

$$R_R = 1 - \frac{R + D}{I} \quad (1.25)$$

Činnost robustního MBR můžeme pozorovat na Obr. 1.10, kde je obdélník se třemi výběžky na jeho obrysu. Vnější dva obdélníky, které jsou šedou barvou vyznačují standardní MBR před a po škálování. Menší obdélník potom vyznačuje výsledné robustní MBR, který rozměrům černému obdélníku s výběžky přibližně odpovídá. Na Obr. 1.11 můžeme vidět další příklady toho, jak robustní MBR pasuje k různým tvarům. Je také vidět, že MBR je schopno celkem spolehlivě pokrýt ty nejdůležitější obdélníkové části daných regionů i přesto, že jsou v jejich tvaru přítomné celkem výrazné odchylky. [19]

### Tvar trojúhelníku

Tuto vlastnost můžeme stejně jako v případě elipticity popsat pomocí momentových invariant. Každý trojúhelník můžeme považovat za jednoduchý pravoúhlý trojúhel-

ník, který je zarovnaný s osami poté, co na něj aplikujeme afinní transformaci. Potom s pomocí vzorce můžeme momenty vyjádřit jako: [19]

$$m_{pq} = \int_0^x \int_0^1 x^p y^q dx dy \quad (1.26)$$

v tomto případě můžeme odvodit  $I_1$  jako: [19]

$$I_1 = \frac{1}{108} \quad (1.27)$$

Potom míra podobnosti s trojúhelníkem  $T_I$  je: [19]

$$T_I = \begin{cases} 108I_1, & \text{if } I_1 \leq \frac{1}{108} \\ \frac{1}{108I_1}, & \text{jinak} \end{cases} \quad (1.28)$$

Míru podobnosti s trojúhelníkem můžeme popsat i dalšími metodami jako aproximace polygonálního trojúhelníku, projekce, přiřazení momentů a další. [19]

## Index formy

Tento index navrhl Masad et al. [1] pro popis ve 2D. Využívá změny přírůstků v poloměru částice. Poloměr označuje délku úsečky, která spojuje střed částice s bodem jejího obrysu. Index formy (dále jen IF) můžeme vyjádřit následující rovnicí: [1]

$$IF = \sum_{\theta=0}^{\theta=360-\Delta\theta} \frac{|R_{\theta+\Delta\theta} - R_{\theta}|}{R_{\theta}} \quad (1.29)$$

Ve vzorci je  $\theta$  směrový úhel a  $R$  je poloměr v různých směrech. Z rovnice vyplývá, že pokud by částice byla dokonalý kruh, tak by vypočítaná hodnota odpovídala číslu 0. [1]

## Úhlovitost

Při měření úhlovitosti se opět omezíme na dvojrozměrné černobílé obrazy. Předpokládáme, že elementy úhlovitosti ve 2D projekcích odpovídají s velkou přesností úhlovitosti ve 3D projekcích. K určení úhlovitosti můžeme využít fourierovskou analýzu. Ta slouží k určení i dalších vlastností, například IF. Úhlovitost s pomocí fourierovi analýzy můžeme zapsat jako: [1]

$$\alpha_r = \sum_{j=5}^{25} [(\frac{a_n}{a_0})^2 + (\frac{b_n}{a_0})^2] \quad (1.30)$$

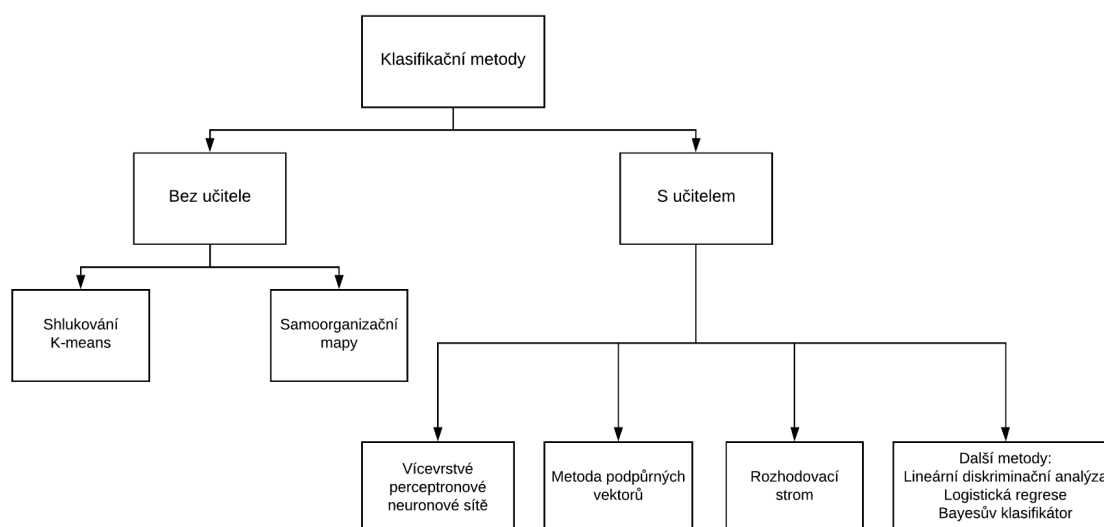
Ve vzorci odpovídají  $a_0, a_n$  a  $b_n$  tzv. fourierovým koeficientům. Ty jsou popsány jinými rovnicemi. Úhlovitost je zaznamenána s pomocí harmonických funkcí s frekvencemi, které jsou vyšší než dané formy, ale nižší než dané struktury. [1]

Popsat by se dalo velké množství dalších deskriptorů, ať už složitějších nebo jednoduchých, nicméně tyto deskriptory uvedené výše slouží jako seznámení s teorií tvarových deskriptorů pro představu, čím se deskriptory zabývají. Mezi další tvary, které by se dali s pomocí deskriptorů popsat patří například čtverec, kosočtverec, mnohoúhelníky atd.

## 1.4 Klasifikace obrazových dat

V této části se podíváme na klasifikační metodu, která je následně využita v praktické části práce. Tvarová analýza by měla v ideálním případě vést k výsledkům ze kterých plyne, že je schopna správně klasifikovat obrazová data. Klasifikační metody hrají důležitou roli v diagnostických a vzdělávacích účelech v medicíně. Existuje velká řada klasifikačních problémů v medicíně, kde bychom se bez tvarové analýzy s pomocí deskriptorů neobešli, protože by je bylo pouhým pohledem obtížné nebo dokonce nemožné jakkoliv řešit. Obecně se při implementaci klasifikátoru v klinickém zpracování obrazu používají klasifikační techniky s učitelem. Vhodnou klasifikační metodou pro naši práci je vícevrstevná dopředná neuronová síť. [11]

Další významné klasifikační metody můžeme vidět na diagramu na Obr. 1.12:



Obr. 1.12: Přehled klasifikačních metod. Zdroj: [11]

### 1.4.1 Vícevrstvá dopředná neuronová síť

Vícevrstvé dopředné neuronové sítě trénované s algoritmem zpětného šíření jsou velmi hojně využívané. Neuronová síť tohoto typu se skládá z neuronů, které jsou uspořádány do vrstev. První vrstva se nazývá vstupní vrstva a poslední vrstva se

nazývá výstupní vrstva. Vrstvy mezi těmito dvěma se nazývají skryté vrstvy. V našem případě se jedná o neuronovou síť, jejíž typ učení je tzv. učení s učitelem. To je takový typ učení, při kterém je neuronové síti předložen vzor. [21]

Pro formální popis neuronů můžeme použít tzv. mapovací funkci, kterou označíme  $r$  a která přiřadí každému neuronu  $i$  podmnožinu  $r(i) \subseteq V$ , která se skládá ze všech předků daného neuronu  $i$ . Podmnožina  $r^{-1}(i) \subseteq V$  se poté skládá ze všech předchůdců daného neuronu  $i$ . Každý neuron v dané vrstvě je spojen se všemi neurony v další vrstvě. Propojení mezi  $i$ -tým a  $j$ -tým neuronem je charakterizován váhovým koeficientem  $w_{ij}$  a  $i$ -tý neuron prahovým koeficientem  $\delta$ . Váhový koeficient odráží stupeň důležitosti daného spojení v neuronové síti. Výstupní hodnota  $i$ -tého neuronu  $x_i$  je určen dle rovnic: [21]

$$x_i = f(\xi_i) \quad (1.31)$$

$$\xi_i = \delta_i + \sum_{j \in r_i^{-1}} w_{ij} x_j \quad (1.32)$$

V těchto vzorcích  $\xi_i$  je potenciál  $i$ -tého neuronu a funkce  $f(\xi_i)$  je tzv. přenosová funkce. Suma ve vzorci (1.32) se provádí na všech neuronech  $j$ , které přenáší signálu do  $i$ -tého neuronu. Prahový koeficient lze chápat jako váhový koeficient spojení s formálně přidaným neuronem  $j$ , kde  $x_j = 1$ . [21]

Pro přenosovou funkci platí: [21]

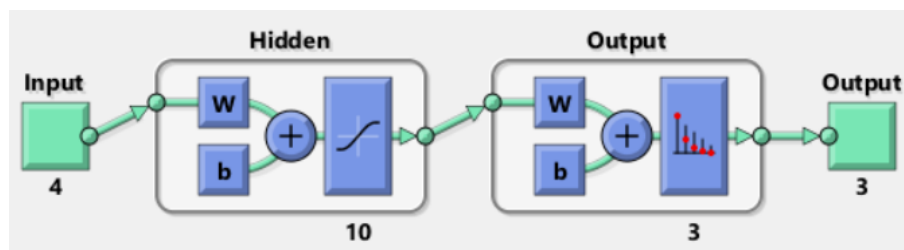
$$f(\xi) = \frac{1}{1 + \exp(-\xi)} \quad (1.33)$$

Proces adaptace s učitelem mění prahové koeficienty  $\delta_i$  a váhové koeficienty  $w_{ij}$ , aby minimalizoval součet umocněných rozdílů mezi vypočítanými a požadovanými výstupními hodnotami. Toho dosáhneme minimalizací funkce  $E$ : [21]

$$E = \sum_o \frac{1}{2} (x_o - \hat{x}_o)^2 \quad (1.34)$$

V tomto vzorci jsou  $x_o$  a  $\hat{x}_o$  vektory složené z vypočítaných a požadovaných aktivit výstupních neuronů a suma probíhá přes všechny výstupní neurony  $o$ . [21]

Na Obr. 1.13 můžeme vidět schéma naší neuronové sítě, kterou budeme používat pro klasifikaci objektů. Jedná se o typ neuronové sítě, která je založená na rozpoznání vzorů. Lze ji tedy trénovat pro klasifikaci vstupů podle cílových tříd. [15]



Obr. 1.13: Vícevrstevná dopředná neuronová síť. Vlevo je počet vstupních proměnných. Následuje skrytá vrstva s deseti neurony. Na výstupu jsou tři klasifikační třídy. Zdroj: [15]

Na schématu v Obr. 1.13 můžeme vidět jako příklad čtyři vstupní parametry a také, že síť obsahuje jednu skrytou vrstvu s deseti neurony. Na výstupu pak můžeme pozorovat, že byly data klasifikována do tří různých tříd. [15]

## 1.4.2 Korelační analýza

Korelační analýza bude v praktické části využita pro správnou klasifikaci dat, a proto je popsána v této kapitole. Tato část tedy vysvětlí základy teorie korelační analýzy, které byly následně využity k lepší funkci algoritmu. Korelační analýza měří sílu vazby mezi dvěma proměnnými a směr této vazby. Sílu vazby vyjadřuje korelační koeficient, který nabývá hodnot od -1 do 1. Hodnota  $\pm 1$  vyjadřuje dokonalý stupeň vazby mezi dvěma proměnnými. Čím více se hodnota korelačního koeficientu blíží nule, tím slabší bude vazba. Směr vazby je vyjádřen znaménkem koeficientu. Tohoto principu využijeme při selekci příznaků do neuronové sítě, kdy budeme vyřazovat příznaky s vysokou vzájemnou korelací. [6]

### Pearsonův korelační koeficient

Pearsonova  $r$  korelace je nejvíce používanou statistickou korelací pro měření síly vazby mezi lineárně závislými proměnnými. Následující vzorec vyjadřuje výpočet Pearsonova korelačního koeficientu: [6]

$$r_{xy} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} \quad (1.35)$$

V tomto vzorci vyjadřuje  $r_{xy}$  Pearsonův korelační koeficient mezi  $x$  a  $y$ ,  $n$  vyjadřuje počet pozorování,  $x_i$  je hodnota  $x$  pro  $i$ -té pozorování a  $y_i$  je hodnota  $y$  pro  $i$ -té pozorování. Pro Pearsonovu korelaci by měli mít obě proměnné normální rozdělení. Hlavním předpokladem je tedy linearita. Linearita předpokládá přímý

vztah mezi každými dvěma proměnnými. Dalším faktorem je poté homoskedasticita, která předpokládá, že data jsou rovnoměrně distribuována kolem regresní linie. [6]

## Studentův $t$ -test

Studentův  $t$ -test je zde popsán za účelem vysvětlení toho, jak byl použit při korelační analýze v praktické části. Obecně je to statistická metoda, ve které testujeme hypotézy o průměru malého vzorku, který byl odebrán z normálně distribuované populace, u které není známá standardní odchylka této populace. [5]

Rozdělení  $t$  je skupina křivek, ve které počet stupňů volnosti (tedy počet nezávislých pozorování ve vzorku mínus jedna) určuje konkrétní křivku. Čím je větší velikost vzorku, tím více se rozdělení  $t$  blíží tvaru zvonu normálního rozdělení. Obvykle se nejdříve formuluje nulová hypotéza  $H_0$ , která říká, že neexistuje žádný efektivní rozdíl mezi pozorovaným průměrem vzorku a předpokládaným nebo uvedeným průměrem populace. Tedy že naměřený rozdíl je způsoben náhodou. [5]

Obecně může být  $t$ -test dvouvýběrový, který uvádí, že průměry nejsou stejné, nebo jednovýběrový, který uvádí, jestli je sledovaný průměr větší nebo menší než předpokládaný. Poté je spočítána testová statistika  $t$ . Pokud je pozorovaná  $t$ -statistika větší než kritická hodnota stanovená vhodným referenčním rozdělením, nulová hypotéza je zamítnuta. Vhodným referenčním rozdělením pro  $t$ -statistiku je  $t$  rozdělení. Kritická hodnota závisí na hladině významnosti testu, která udává pravděpodobnost chybného zamítnutí nulové hypotézy. [5]

Tento popsaný princip je v praktické části využit pro výběr vhodných příznaků do neuronové sítě. Statistiku  $t$  jsme v našem případě spočítali z korelační matice podle následujícího vzorce: [18]

$$t = r \sqrt{\frac{n-2}{1-r^2}} \quad (1.36)$$

V tomto vzorci odpovídá  $t$  testové statistice  $t$ ,  $n$  odpovídá počtu stupňů volnosti a  $r$  odpovídá hodnotám korelační matice. [18]

## 1.5 Aplikace tvarové analýzy

V této části se podíváme na konkrétní využití tvarové analýzy. Jelikož její využití je velmi široké a její aplikace můžeme pozorovat v různých oblastech ať už medicíny nebo jiných, uvedeme zde pouze několik příkladů. Její využití je ovšem mnohem širší, nicméně cílem této části práce je poukázat na její výhody, dostupnost a praktičnost.

Uvedené příklady využití jsou pokud možno co nejvíce aktuální, abychom opravdu nastínili její současné využití. Tyto příklady také budou obsahovat i jiné metody a

postupy tvarové analýzy, než které jsme popsali výše. Zejména proto, že v praxi se většinou využívá mnohem více metod na jednou a také proto, že se tvarová analýza aplikuje i na 3D data na rozdíl od této práce.

### 1.5.1 Tvarová analýza mozkových komor s pomocí SPHARM

V této části se podíváme na metodu, která je původně vynalezena pro segmentaci založenou na modelech, která se nazývá SPHARM (*spherical harmonics* - sférické harmonické). Tuto metodu, která slouží k reprezentaci tvaru povrchu různých objektů, použijeme pro analýzu mozkových struktur. Obzvlášť se v této aplikaci metoda zaměřuje na problém klinického výzkumu a to studia podobnosti mozkových struktur v indetických (monozygotních) a neidentických (dizygotních) dvojčat. [9]

Pokud se pokusíme nějak obecně shrnout metodu SPHARM, jedná se o hierarchický, globální, multi-škálovací popis obrysu, který může reprezentovat pouze objekty sférické topologie. Základní funkce parametrizovaného povrchu jsou sférické harmonické frekvence. Je dokázáno, že SPHARM může být použit k vyjádření deformací tvaru. Zkrácením sférické harmonické série pod různými stupni dosáhneme výsledku ve formě reprezentací objektu s různou úrovní detailu. SPHARM je velmi citlivá, přesně škálovaná reprezentace tvaru. Samozřejmě počítáme s vhodně malou chybou aproximace. [9]

Důležité potom je, že korespondence mezi objekty popsány metodou SPHARM nám umožňuje výpočet vzdálenosti mezi dvěma objekty. Ortogonalita sférických harmonických základních funkcí je následně využita k výpočtu střední kvadratické vzdálenosti ( $\sqrt{MSD}$ ) mezi dvěma objekty přímo z jejich koeficientů. Je také potřeba korekce, protože odmocněné sférické harmonické funkce neintegrují na hodnotu 1, ale na hodnotu  $4\pi$ . [9]

### 1.5.2 Detekce plicních uzlin v CT obrazech hrudníku s pomocí tvarové analýzy

V uplynulých letech proběhl velmi výrazný rozvoj výpočetní tomografie (dále jen CT) jako nástroje pro detekci nádorů plic. Plicní uzliny mohou být rozděleny na dvě kategorie. První kategorie jsou pevné (*solid*) uzliny charakteristické jejich vysokým kontrastem. Druhá kategorie jsou potom tzv. *ground-glass opacity* (dále jen GGO) uzliny se slabým kontrastem a rozmazanými okraji. Tyto GGO uzliny jsou s větší pravděpodobností zhoubné než pevné uzliny. Studie na téma detekce plicních uzlin s pomocí počítače se hojně vyskytují v literatuře, ale největší pozornost byla obrácena na detekci pevných uzlin. [26]

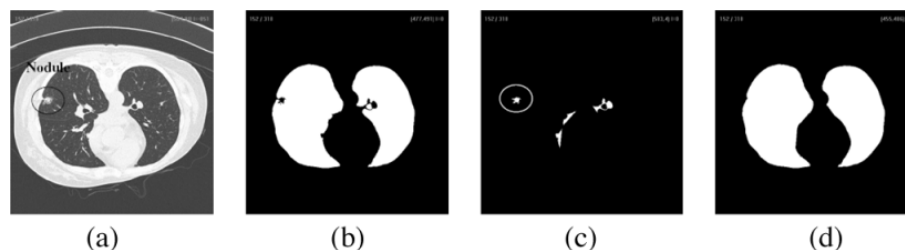
Existuje také několik článků, které popisují detekci GGO uzlin, protože ta by podle většiny odborníků měla být hlavní prioritou detekce v této oblasti. Většina tohoto výzkumu se soustředí na detekci pomocí CT s vysokou dávkou. Pro zobrazení plicních nádorů se ovšem ukázalo jako efektivní i CT o nízké dávce. Další metodou, kterou si blíže popíšeme je metoda CT plic CAD (*computer-aided* - s pomocí počítače), jejíž cílem je detekovat jak pevné uzliny, tak i GGO uzliny. [26]

Tato metoda je relativně nový a efektivní přístup k CAD plic díky výpočtu 3D lokální geometrie a statistické intenzity pro potenciální detekci pevné a GGO uzliny. Jejím cílem je vytvořit sjednocenou strukturu plicního CAD s klinicky přijatelným výkonem pro detekci pevných i GGO uzlin a to s použitím širokého rozsahu různých zkumavek (většinou v rozsahu 30-250 mA). [26]

Samotný algoritmus plicního CAD je složen z několika známých metod, které jsou ovšem pro tuto metodu náležitě upraveny. Mezi tyto metody patří především následující. Nejdříve se aplikuje adaptivní neostré prahování pro segmentaci regionu plic. Následně se aplikuje antigeometrický rozptyl na obraz předtím, než vypočítáme geometrické vlastnosti. Hladkost hran generovaných antigeometrickým rozptylem je vhodnější pro přesný výpočet vlastností založených na voxelech. Kombinace vlastností je vhodná pro detekci pevných a GGO uzlin. Mezi dobré deskriptory struktury, které jsou schopné správně detekovat většinu uzlin, patří například index tvaru, který udává informaci o tvaru dané oblasti. Správná segmentace objektu (uzliny) je velmi důležitá pro získání vlastností. Řešení pro efektivní segmentaci objektu, který je potenciální uzlinou, se skládá ze dvou kroků. Prvním krokem je tedy již zmíněné adaptivní prahování pro hrubou segmentaci a druhým krokem poté modifikovaný algoritmus (tzv. očekávané maximalizace) pro přesnější segmentaci. [26]

V článku, který pojednává o této problematice, je uvedeno, že tato metoda byla zhodnocena na klinickém datovém souboru 108 hrudních CT snímků s rozsahem zkumavek od 30 do 250 mA. Výsledky experimentu poukazují na vysoký výkon této metody s přesností detekce kolem 90,2 procent a to jak pro pevné, tak i pro GGO uzliny. [26] Tato metoda se sice pohybuje v oblasti 3D obrazových dat, kterou se naše práce nezabývá, ovšem příklad slouží jako názorná ukázka využití tvarové analýzy v oblasti medicínských dat, konkrétně využití CT snímků, se kterými budeme později pracovat. [26]





Obr. 1.14: Segmentace plic s pomocí metody neostrého prahování. (a) Původní obraz z CT plic; (b) obraz po prahování; (c) segmentované konkávní regiony (připojená uzlina a další tkáně); (d) výsledný segmentovaný obraz na základě (b) a (c). Zdroj: [26]

### 1.5.3 Tvarová analýza mamografických kalcifikací

Nejenže je rakovina prsu hlavní příčinou smrti rakovinou u žen středního a pokročilého věku, ale její četnost stále roste. Prevence a včasná diagnóza jsou tedy velmi důležité. Samozřejmě příčina této nemoci stále ještě není úplně jasná. Ovšem současné metody léčby jsou velmi efektivní v počáteční fázi onemocnění. Mamografie je jediná zobrazovací metoda schopná detekovat rakovinu prsu v její počáteční fázi a to co nejdříve. Výsledky mamogramu jsou zhodnoceny experty, kteří vizuálně vyhodnotí přítomnost deformací, které by mohly reprezentovat začínající rakovinu. V současnosti se samozřejmě i v tomto případě nachází různé limitace ohledně zobrazení počítačem. Nicméně s pomocí počítačové technologie můžeme dosáhnout automatické mamografické analýzy. Důkaz přítomnosti rakoviny prsu je obvykle indikován obsahem mamogramu obecně nebo pozorujeme působení rakoviny na její okolí. Hlavním znakem malignity na mamografickém snímku jsou seskupené kalcifikace, špatně definované hmoty, architektonické zkreslení, vývojová hustota, asymetrie a další. [14]

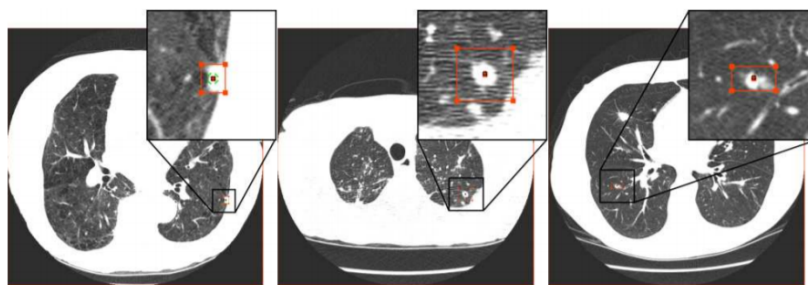
Vzhledem k tomu, že některé tyto vlastnosti se špatně pozorují na mamogramu, zvláště při počáteční fázi onemocnění, existuje velké množství prací, zabývajících se zlepšením vlastností mamografie. Mezi ně patří například filtrování s pomocí tzv. neostré masky v digitální mamografii. Dále také algoritmy založené na adaptování sousedním pixelům ve zpracování obrazů. Ohledně kalcifikací existují například technologie pro obdržení vhodně zvětšených obrazů kvůli ulehčení diagnózy kalcifikací a pro zvětšení její přesnosti. Wee et al. [14] vynalezl proceduru, která využívá soubor několika vlastností. Mezi ně patří plocha, střední úroveň šedi, kontrast, odchylka úrovně šedi, horizontální a vertikální délka, kvantita tvaru a další pro odlišení kalcifikací na mamogramu. Spiesberger [14] navrhl podobnou proceduru s odlišným souborem vlastností pro nalezení kalcifikací. Davies a Dance [14] zkoumali segmentaci pro detekci nashromážděných kalcifikací založených na procesu lokálního prahování. [14]

Feig et al. a Sickles [14] zjistili, že jedním z hlavních kritérií pro odlišení maligních kalcifikací od benigních kalcifikací je jejich tvar. Již předtím se využívali standardní metody jako výpočet vzdáleností bodů kontury od těžiště regionu nějaké kalcifikace. Ovšem tato metoda nedokáže popsat přesnou ostrost tvarů. Spiesberger [14] využil u měření typu kompaktnosti, aby charakterizoval rozptýlenou či kompaktní povahu kalcifikací. Toho se využilo již dříve, aby se odlišili kalcifikace od normálních prsních struktur. Mezi další tvarové deskriptory, které se využili, patří například obsah, excentricita, koeficient konvexnosti, rychlost prodlužování. Ovšem tyto vlastnosti, až na kompaktnost, nemusí přímo charakterizovat ostrost tvaru kalcifikací. [14]

#### **1.5.4 Modelová analýza lokálního tvaru pro detekci lézí v CT skenech**

Nejmodernější snímače počítačové tomografie, které mají menší rozlišení než jeden milimetr, poskytují obrazy stále menších uzlin, které jsou charakteristické pro počáteční stav rakoviny plic. Ovšem toto zlepšení prostorového rozlišení vede také k tomu, že odborníci musí přezkoumat mnohem více obsahu v obrazech, z čehož plyne mnohem větší variabilita interpretace daného snímku. [16]

Na Obr. 1.15 můžeme vidět detekované uzliny dvěma radiology, ale třetí radiolog je nezaznamenal. Detekce s pomocí počítače byly vytvořeny, aby radiologům pomohli číst správně snímky CT. Brown et al. [16] použil jednoduchý fuzzy klasifikátor založený na vlastnostech tvaru a intenzity segmentovaných regionů. Data se skládali pouze z vybraných průřezů plic o šířce dvou centimetrů a upřednostňoval 3D segmentační techniku, na které je algoritmus závislý a vyhýbá se složité oblasti plicních apexů. McCulloch et al. [16] obdržel dobré výsledky s využitím Bayesova klasifikátoru - 70% senzitivita při 8 falešných pozitivních na jeden případ s lehce zašuměnými screenovacími daty. Metoda používá 2D segmentaci pro generování kandidátů a počet falešně pozitivních výsledků roste s použitím tenčích řezů CT. [16]



Obr. 1.15: Tři příklady struktur, které nebyly zaznamenány radiologem, ale nezávisle na tom klasifikovány jako uzliny dalšími dvěma radiology. Tyto abnormality byly úspěšně detekovány algoritmem navrženým v článku. Zdroj: [16]

Paik et al. [16] použil geometrický model a učící parametry pro trénovací data. Poukazují na výsledky pouze osmi CT řezů pro uzliny s průměrem nad 6 mm, tedy z toho nemohou být vyvedeny žádné solidní výsledky. Farag et al. [16] došel k výsledkům až 80,3% senzitivity s 9,2% falešně pozitivními. Ovšem většina těchto uzlin byla kalcifikována, a tudíž jsou klinicky irelevantní. Přes 80% se skládá z uzlin větších než 12 mm a ty jsou lehce detekovány. Byli použity i další metody založené na lokální strukturní informaci pro rozlišení mezi jednotlivými tvary, ale ve všech těchto metodách byly prahy pro klasifikaci buď nastaveny empiricky nebo naučeny z dat. [16]

Metoda pro detekci lézí je založena na 3D algoritmu, který nezávisí na segmentaci nebo shlukování relevantních anatomických struktur nebo na jakékoliv detekci hran. Navrhovaný algoritmus kombinuje geometrické modely a modely založené na intenzitě, s vlastní analýzou tenzoru zakřivení za účelem identifikace plicních uzlin v CT. Všechny parametry a prahy jsou derivovány z těchto modelů, čímž se eliminuje potřeba trénovacích dat. Navíc není provedena žádná optimalizace. Všechny operace jsou lokální, a tudíž navrhovaná metoda může být interpretována jako filtr pro zvýraznění uzlinám podobných struktur v obraze. [16]

### Další příklady aplikace

- Tvarová analýza hipokampu u schizofrenie [20]
- Porovnání sledě otolitů s použitím tvarové analýzy [3]
- Charakterizace tvaru částic pomocí fourierova deskriptoru [4]
- Porovnání tvarových sekvencí ve videu s aplikacemi analýzy lidského pohybu [25]
- Eliptická fourierova tvarová analýza fosilních mlžů [8]

## 2 Praktická část

V této části studentské práce se budeme zabývat využitím metody tvarové analýzy na konkrétních datech. K tomu nám budou sloužit metody a postupy, které jsme popsali výše v kapitole teoretická část. Pro veškeré výpočty a aplikace na konkrétní data byl využit jazyk MATLAB a to zejména kvůli jeho široké oblasti použití, komunitě a také jeho přednostem ve zpracování obrazu.

Nejdříve se zaměříme na implementaci popsané teorie o tvarové analýze, abychom dosáhli požadovaných výsledků v praktické části. V další části budou postupy následovně aplikovány na jednoduchá dvojrozměrná data, která jsme zvolili. Dále se poté budeme zabývat implementací našeho algoritmu na reálná data z medicínské databáze. Výsledkem práce bude aplikování tvarové analýzy pro klasifikaci objektů v medicínských obrazech, v našem případě se konkrétně budeme zabývat klasifikací obratlů na lumbální a thorakální.

### 2.1 Aplikace tvarové analýzy na základní 2D obrazy

V této části se budeme zabývat aplikací metod, které jsme popsali v teoretické části v kapitolách 1.2, 1.1, 1.3, na konkrétní data. Jak jsme již psali výše, budeme k tomu využívat různé obrazy, ať už vygenerované nebo získané z externích zdrojů. Obecně budeme v algoritmech pracovat s binárními obrazy, jelikož se v algoritmech zabýváme především tvarovou analýzou a klasifikací, tedy nám binární obrazy postačí.

Samotný algoritmus našeho řešení popíšeme v těchto podkapitolách:

1. Použitá obrazová data
2. Předzpracování obrazových dat
3. Aplikace tvarových deskriptorů
4. Klasifikace objektů

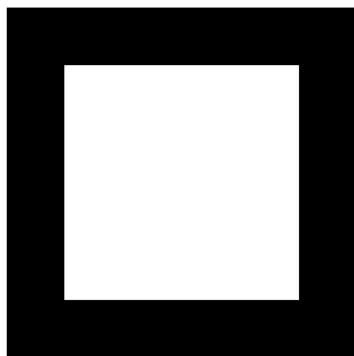
#### 2.1.1 Použitá obrazová data

Nejdříve je vždy nutné do našeho programovacího prostředí načíst obrazy, se kterými chceme dále pracovat. Jako vstupní data do našeho algoritmu jsme použili především námi vygenerované geometrické objekty. Jejich výhodou je, že jich můžeme jednoduše vygenerovat velký počet o různých parametrech. Velké množství těchto vygenerovaných obrazů pak bude dobře sloužit jako statistické vyhodnocení úspěšnosti našeho algoritmu.

V programovacím prostředí jsme vytvořili několik algoritmů, které s pomocí jednoduché práce s maticemi generují různé geometrické objekty. Například pro jednoduchý algoritmus na vytvoření čtverce jsme využili základního vzorce  $a * a$  pro výpo-

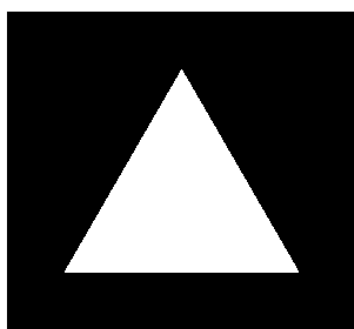
čet obsahu. Pro vytvoření binárního obrazu byl tento obsah matice zaplněn hodnotami 1. Na podobném principu jsme vytvořili i ostatní geometrické objekty pro kruh, obdélník, trojúhelník, kosočtverec a elipsu.

Na Obr. 2.1 můžeme vidět příklad vygenerovaného čtverce v programovacím prostředí. Takových čtverců bylo vytvořeno velký počet vždy o náhodných parametrech, abychom prokázali funkci algoritmu na co největší počet různých dat. V tomto případě byla zadaná strana čtverce tvořena 200 pixely.



Obr. 2.1: Čtverec vytvořený v programovacím prostředí

Pro názornou ukázkou je ještě na Obr. 2.2 zobrazen vygenerovaný trojúhelník. Stejně jako v případě čtverce a ostatních objektů odpovídají hodnoty 1 obsahu geometrického objektu a hodnoty 0 pozadí.



Obr. 2.2: Trojúhelník vytvořený v programovacím prostředí

## 2.1.2 Předzpracování obrazových dat

Předtím než budeme na data implementovat metody tvarové analýzy, je potřeba tato data předzpracovat k dalšímu použití. V rámci tvarové analýzy bylo důležité převést naše načtené obrazy z barevných na binární, protože v rámci tvarové analýzy se s nimi dále lépe pracuje a navíc nás v tomto případě zajímá pouze jakási hlavní maska objektů.

### Binarizace

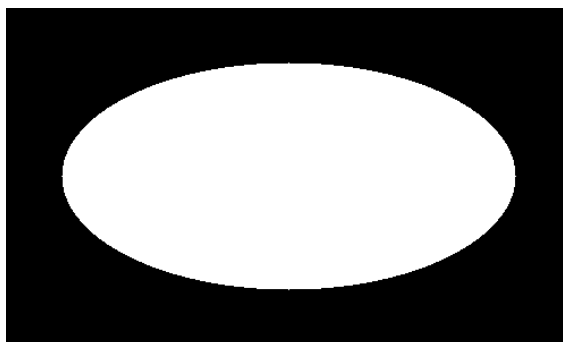
Binarizace je základní operací pro práci s obrazy, při které je obraz zjednodušen na černobílý (binární). Nejdříve jsme tedy vždy načetli obraz, a následně ho převedli na binární pro správnou funkci algoritmu s pomocí jednoduchého prahování, kde hodnoty 0 odpovídají černé barvě (pozadí) a hodnoty 1 odpovídají barvě bílé (objektu). Vygenerované obrazy uvedené výše už jsou při vytvoření binární. Pro správnou funkci našeho algoritmu a zejména pro přehlednost jsme nastavili obrazy tedy vždy tak, aby byl samotný objekt složený z hodnot 1, tedy objekt se jeví jako bílý, zatímco pozadí bylo složené z hodnot 0, tedy se pozadí jeví jako černé. Z toho plyne, že samotné objekty obsahují hodnoty 1 také proto, abychom u nich mohli spočítat například obsah.

### Získání obrysu ve 2D obrazech

Převedení na binární obraz je první a základní operace. K tomu, abychom obraz mohli správně klasifikovat, ale musíme použít další pokročilejší metody, které byly zmíněny výše. Jako první na obrazová data použijeme metodu segmentace obrazu popsanou v kapitole 1.2 a to detekci hran, jejíž cílem je segmentovat obraz a zobrazit jeho objekty, přičemž detekuje obrys těchto objektů. Jak jsme se již dozvěděli z kapitol teorie, můžeme definovat hranu jako křivku v obraze, která sleduje prudké změny intenzity.

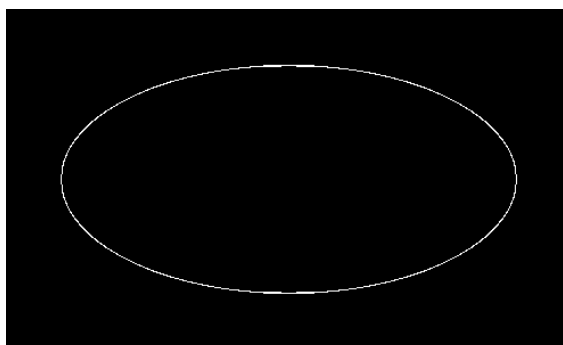
Pro nalezení obrysu jsme využili funkce programovacího prostředí. V obraze jsme vyhledali místa, kde se prudce mění intenzita. K tomu se využívá jednou ze dvou kritérií. Buď se jedná o místo, kde první derivace intenzity je větší než daný práh nebo se jedná o místo, kde druhá derivace intenzity má tzv. nulový bod, což je místo, kde se mění její znaménko. Díky tomu jsme dostali obraz obsahující hodnoty 1 na místech, kde byl nalezen obrys, a hodnoty 0 všude jinde.

Pro názornější ukázkou jsme jako vstupní obraz použili vygenerovanou elipsu na Obr. 2.3, na kterém byl algoritmus testován a na kterém můžeme získání obrysu dobře pozorovat.



Obr. 2.3: Vygenerovaný obraz elipsy.

Nejvíce efektivní metodou získání obrysu, kterou náš algoritmus umožňuje je tzv. Cannyho detektor, díky kterému je naše metoda méně ovlivněna šumem a více pravděpodobně bude detekovat správně obrys. Náš algoritmus tedy ukazuje, jak detekovat obrys v obrazu s pomocí Cannyho detektoru a na Obr. 2.4 můžeme pozorovat, jak takový obraz po nalezení obrysu vypadá.



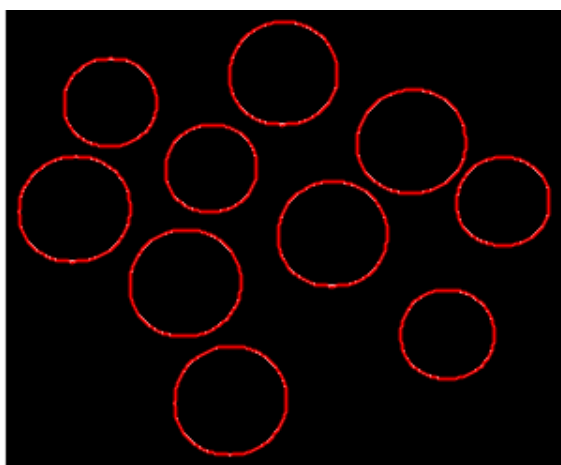
Obr. 2.4: Získaný obrys vygenerované elipsy s pomocí Cannyho detektoru.

Z Obr. 2.4 tedy můžeme vidět, že Cannyho detektorem se nám povedlo úspěšně získat obrys geometrického objektu. Takto bylo možné získat obrys i všech ostatních geometrických objektů, které jsme generovali. V tomto případě, kdy budeme pracovat nejdříve s geometrickými objekty, nám získání obrysu Cannyho detektorem dostačuje, protože skutečně detekuje pouze konturu objektu a nic jiného.

### **Obrysové trasování ve 2D obrazech**

Obrysové trasování ve 2D obrazech funguje na principu, který jsme popsali v teoretické části v kapitole 1.2. Pro ukázkou aplikace jsme tentokrát použili obraz obsahující mince z externího zdroje. Vytvořený algoritmus je schopen detekovat startovní body trasování a poté označit průběh trasování na objektech v obrazu.

Algoritmus dále funguje tak, že nalezne souřadnice počátečního bodu obrysu objektu, odkud bude následně zahájeno trasování. K obrysovému trasování samotnému potom slouží funkce programovacího prostředí, která vrátí souřadnice všech pixelů obrysu daného objektu v obraze. Kromě správného nastavení souřadnic startovního bodu trasování bylo pro algoritmus také důležité určit směr trasování. Následně byl také algoritmus upraven tak, aby byl schopen provést obrysové trasování pro více objektů v obraze. Při vkládání různých obrazů se může stát, že vstupní binární obraz není rozdělený bílou a černou barvou pouze na pozadí a objekty, ale může se například hodnota 0 nalézat i uvnitř objektu a algoritmus by tedy rozpoznal tyto hodnoty jako samostatné objekty. Abychom docílili toho, že budeme trasovat pouze objekty, které nás zajímají, bylo tedy nutné větší objekty zájmu zaplnit hodnotami 1, a potom se tedy už uvnitř nenacházely žádné objekty menší. Toto se zejména hodí ve chvíli, kdy budeme pracovat s medicínskými daty a budeme chtít zanedbat menší nepodstatné objekty, které se v obrazech nachází. A to zejména kvůli tomu, že nás zajímá především vnější charakteristika objektů.



Obr. 2.5: Obraz s mincemi po aplikaci obrysového trasování. Červenou barvou je vyznačen obrys mincí.

Stejně jako u elipsy jsme nejdříve našli konturu všech objektů v obraze. Na Obr. 2.5 můžeme pozorovat vykreslení obrysového trasování v obraze na všech objektech, kde červená barva vyznačuje nalezený obrys. Podle přiloženého obrázku můžeme pozorovat, že jsme byli v trasování obrysu úspěšní, protože všechny objekty v obraze byly správně zaznamenány a to i jejich obrys.

### 2.1.3 Aplikace tvarových deskriptorů

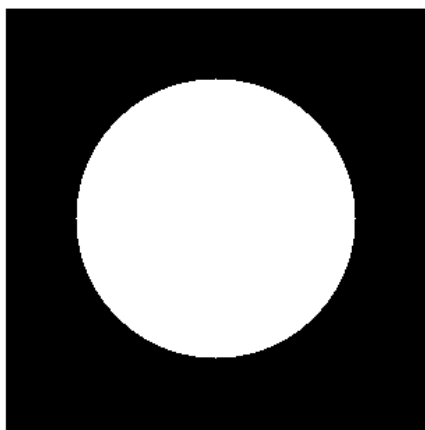
Po předešlé přípravě obrazových dat a otestování metod segmentace obrazu můžeme na naše vstupní data aplikovat tvarové deskriptory. V této části se tedy podíváme



na rozpoznání jednotlivých geometrických tvarů v obrazech v našem programovacím prostředí. Několik tvarových deskriptorů, které můžeme aplikovat do programovacího prostředí, bylo popsáno v kapitole 1.3.4. K určení a klasifikaci tvaru nějakého objektu potřebujeme algoritmus, který získá vlastnosti objektu a na jejich základě určí výsledný tvar. Mezi takové vlastnosti patří obvod, obsah a další specifické vlastnosti, které od sebe jednotlivé tvary oddělí. V našem řešení použijeme pro rozlišení objektů zejména tyto parametry: velké množství geometrických deskriptorů, které jsou implementovány přímo v programovacím prostředí, fourierovské deskriptory a kontury.

### **Získání geometrických deskriptorů**

V této části si vysvětlíme princip získání různých deskriptorů z objektů. Nejdříve se budeme zabývat geometrickými deskriptory. Mezi jeden z nejdůležitějších tvarů, který můžeme rozpoznat, patří tvar kruhu, případně míra kruhovitosti objektu. Toto slouží pouze jako jednoduchý názorný příklad výpočtu tohoto deskriptoru pro několik objektů, abychom poukázali na princip získání deskriptorů. Pro názorný příklad uvedeme vygenerovaný kruh na Obr. 2.6, který odpovídá téměř ideálnímu kruhu.



Obr. 2.6: Vygenerovaný obraz kruhu.

K výpočtu míry kruhovitosti jsme využili přímo programovacího prostředí, nicméně mohli bychom například také použít vzorec (1.20) uvedený v kapitole 1.3.4. I v tomto případě jsme nejdříve získali obrys kruhu. Následně jsme v programovacím prostředí získali soubor vlastností objektu, se kterým pracujeme. Díky získaným vlastnostem

kruhu poté můžeme snadno dohledat vypočítanou míru kruhovitosti. Ta by měla v ideálním případě vycházet 1.

Díky tomuto algoritmu zjišťujeme, že výsledná míra kruhovitosti vychází 1,004 i přesto, že obraz, který byl použit, skutečně byl kruh. Nicméně hodnota po zaokrouhlení na tři desetinná místa skutečně vychází 1, což se dá považovat za velmi přesný výsledek. Takto malé odchylky od hodnoty 1 by se dali vyřešit například větším rozlišením obrazu vygenerovaného kruhu. Pokud bychom si například nastavili klasifikační kritérium pro kruh v rozmezí hodnot 0,95-1,05, tento objekt by byl skutečně jako kruh zařazen. Na tomto principu by mohli fungovat i například klasifikace založené na nějakém souboru podmínek.

Pokud bychom tedy chtěli získat míru kruhovitosti pro další vygenerované obrazy, hodnota bude vycházet jiná než 1. Při měření kruhovitosti dalších generovaných obrazů, kde jejich rozlišení odpovídalo 200 pixelům, nám vycházeli tyto hodnoty:

č.	Tvar obrazu	Vypočítaná hodnota
1	Kruh	1,004
2	Elipsa	0,843
3	Obdélník	0,738
4	Kosočtverec	0,577
5	Čtverec	0,827
6	Trojúhelník	0,630

Tab. 2.1: Tabulka s vypočítanými hodnotami kruhovitosti vygenerovaných tvarů s rozlišením 200px

V Tab. 2.1 můžeme vidět vypočítané hodnoty kruhovitosti pro ostatní generované geometrické objekty. Tabulka slouží jako ukázka toho, že můžeme vyčíst rozdíly ve vypočítané hodnotě pro různé tvary. Kosočtverec má v tomto případě hodnotu nejvíce vzdálenou od ideálního kruhu, tedy od hodnoty 1.

Tímto způsobem jsme získali i další deskriptory a to přímo v programovacím prostředí tak, že jsme tedy získali velký soubor vlastností o každém objektu. Následně ze souboru vlastností vybereme takové vlastnosti, které jsou více specifické pro daný geometrický objekt a které od sebe objekty lépe odliší. Pro lepší specifikaci pak používáme například poměr určitých vlastností. Mezi další deskriptory, které jsme testovali při klasifikaci objektů patřily tyto:

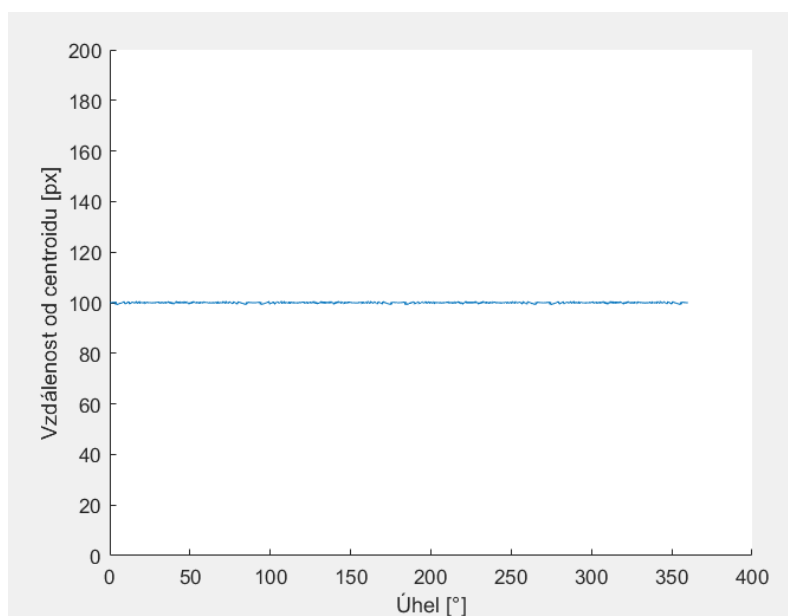
- Obsah/obvod
- Délka hlavní osy/ délka vedlejší osy
- Excentricita

## Získání fourierovských deskriptorů

Mezi další deskriptory, které byly v algoritmu získány pro lepší odlišení jednotlivých tvarů, patří fourierovské deskriptory. Pro jejich aplikaci jsem museli nejdříve získat vhodné vlastnosti objektů, zejména vzdálenost jednotlivých bodů obrysu od centroidu. Nejdříve tedy vytvoříme souřadnicový prostor a získáme souřadnice kontury. Ty poté vektorizujeme a jsou odmazány všechny nulové hodnoty. Následně upravíme hodnoty tak, že odečteme souřadnice x a y centroidu. Hodnoty jsou převedeny do polárních souřadnic a úhly jsou seřazeny od nejmenšího po největší. Následují ještě další úpravy jako například odstranění duplicitních hodnot. Výsledné spektrum dostaneme po fourierově transformaci vzdálenosti obrysu od středu centroidu. Pro lepší klasifikaci objektů bylo tedy potřeba zejména vypočítat vzdálenosti ne jen pro jednotlivé body, ale také pro úhel. Stejně jako v případě geometrických deskriptorů bylo ale i zde využito pro lepší specifikaci při testování například:

- Relativní zastoupení složek v dolní desetinné spektra
- Střední frekvence
- Rozptyl vzdálenosti obrysu od centroidu/ střední hodnota vzdálenosti obrysu od centroidu

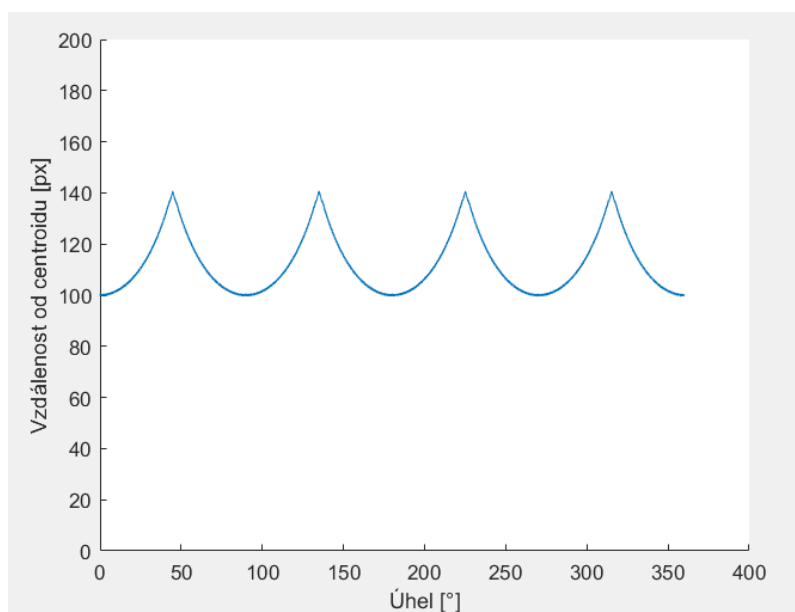
Nyní se podíváme na několik příkladů generovaných geometrických objektů a budeme sledovat, jak vypadá jejich křivka vzdálenosti bodů obrysu od centroidu daného generovaného objektu. Na Obr. 2.7 můžeme vidět, jak tato křivka vypadá pro generovaný kruh.



Obr. 2.7: Vzdálenost bodů obrysu od centroidu u kruhu

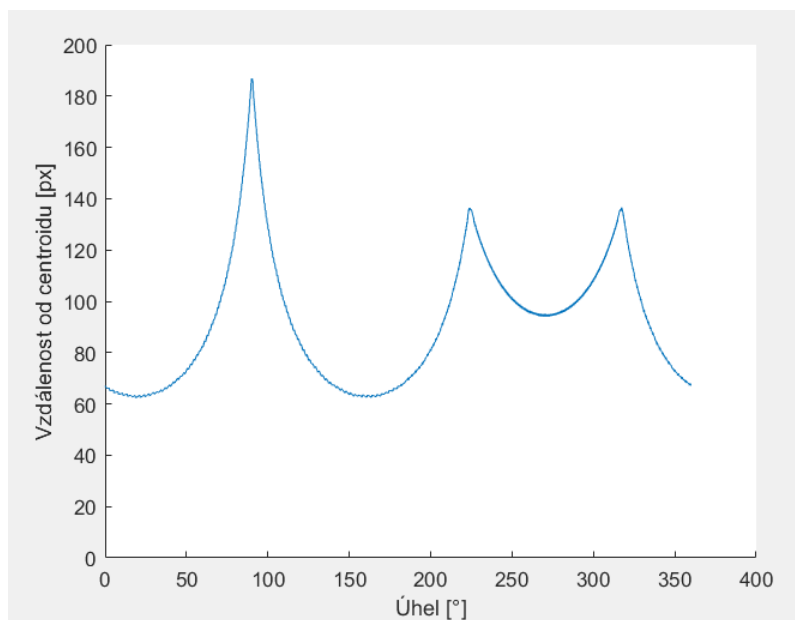
Na dalším Obr. 2.8 můžeme pozorovat, jak vypadá křivka vzdálenosti bodů ob-

rysu od centroidu u generovaného čtverce.



Obr. 2.8: Vzdálenost bodů obrysu od centroidu u čtverce

A pro názorný příklad ještě uvedeme na Obr. 2.9, jak by tato křivka vypadala u generovaného trojúhelníku (v tomto případě rovnoramenný).



Obr. 2.9: Vzdálenost bodů obrysu od centroidu u trojúhelníku

Z výše uvedených příkladů vyplývá, že vzdálenost bodů obrysu o centroidu se pochopitelně mezi jednotlivými objekty velmi liší. Tedy budou deskriptory založené

na tomto principu velmi dobře přispívat klasifikaci geometrických objektů.

## 2.1.4 Klasifikace objektů

V této části algoritmu už dochází k samotné klasifikaci objektů obrazových dat. Jako příklad zde budou uvedeny další objekty, které byly testovány pro rozpoznání správného tvaru a budeme sledovat, jestli byly klasifikovány správně. Především nás ale také zajímá statistická úspěšnost algoritmu, a proto otestujeme algoritmus na velké množství vygenerovaných obrazů různých geometrických tvarů a zjistíme, jak vysoká je úspěšnost detekce.

### Rozpoznání geometrických tvarů ve 2D obrazech

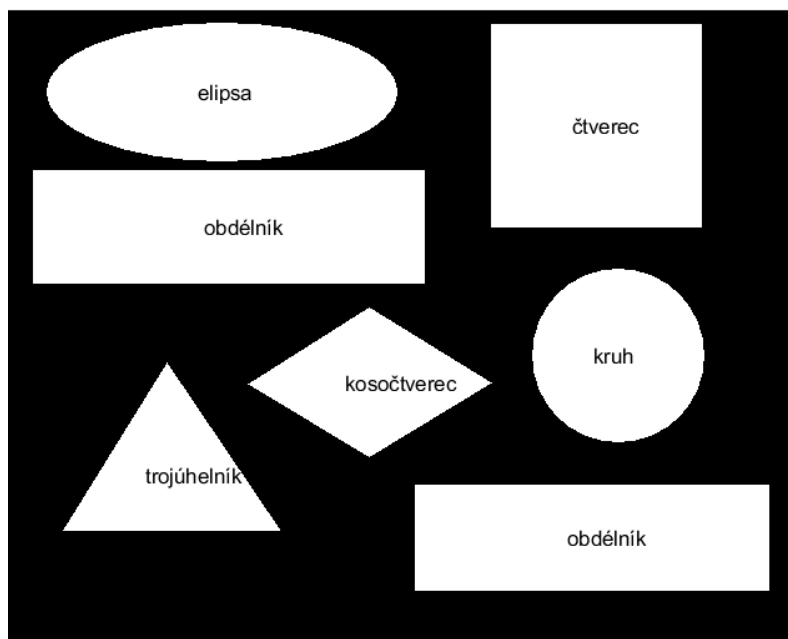
Další část našeho algoritmu slouží k rozpoznání geometrických tvarů od sebe a jejich klasifikaci, tedy tvarů jako je čtverec, trojúhelník, obdélník a další. Dokážeme v obraze jednak klasifikovat tvar jednoho obrazce, pokud je načtený obrázek s jedním objektem, ale také dokážeme rozpoznat více tvarů v obraze najednou. Jako příklad uvedeme rovnou obraz s více tvary, z čehož plyne, že algoritmus se chová univerzálně pro libovolný počet objektů v obraze.

### Rozpoznání objektů na základě minimální vzdálenosti obrysu od těžiště

Stejně jako v předešlých případech klasifikace, které jsme již uváděli, je opět důležité získání charakteristických vlastností jednotlivých objektů. Algoritmus tedy rozpozná jednotlivé tvary a přiřadí název do obrazu k danému objektu. To provede tak, že postupně spočítá pro každý objekt v obraze hodnotu pro míru kruhovitosti, elipticity a další tvary, a poté k němu přiřadí takový tvar, u kterého byla hodnota nejbližší k ideální hodnotě. Tento příklad klasifikace tedy funguje na systému zadaných podmínek. Hlavním principem algoritmu je, že spočítá vzdálenosti souřadnic obrysu od jeho těžiště. Důležité je poté, jestli jsou vypočítané vzdálenosti například všude stejné (v tom případě by se jednalo o kruh) nebo ne. Pokud si vezmeme například čtverec, tak po výpočtu vzdáleností jeho bodů obrysu od těžiště můžeme spočítat obsah objektu. Pokud se vypočítaný obsah objektu přímo rovná vzorci pro výpočet obsahu čtverce (tedy bude v nějakém daném rozmezí kolem hodnoty 1), je objekt klasifikován jako čtverec. Výpočet konkrétně proběhl tak, že algoritmus získal minimální vzdálenost obrysu čtverce od těžiště, kterou si označíme  $d$ . Následně s pomocí vzorce (2.1) tuto vzdálenost umocnil na druhou a po vynásobení číslem 4 jsme získali obsah objektu  $S$ .

$$S = 4d^2 \tag{2.1}$$

Pokud tento vypočítaný obsah souhlasí se získaným obsahem objektu ze souboru vlastností, jedná se o čtverec.



Obr. 2.10: Binární obraz s několika klasifikovanými tvary najednou.

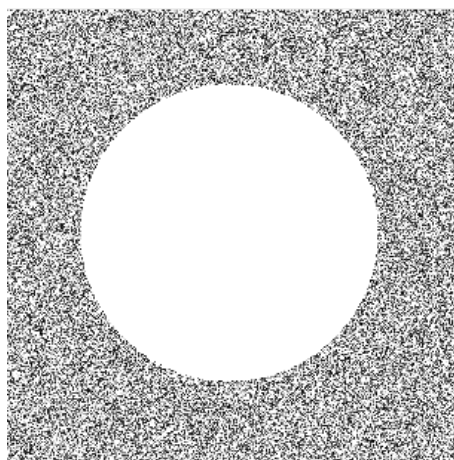
Na dalším Obr. 2.10 se můžeme přesvědčit, že algoritmus tedy dokáže detekovat více objektů v obrazu najednou. Všechny tvary byly klasifikovány správně. U každého tvaru je napsaný název geometrického objektu, který algoritmus přiřadil. Obraz obsahuje všechny geometrické tvary, které jsme v programovacím prostředí generovali, ovšem tentokrát je vstupní obraz z externího zdroje, aby obsahoval všechny tvary najednou a také abychom algoritmus otestovali na jiném náhodném obrazu a ne pouze na našich generovaných objektech. Pro účely vepsání klasifikace tvarů jsou tentokrát objekty vyplněné hodnotami 1.

### Rozpoznání vygenerovaných 2D obrazů s pomocí klasifikátoru

Poté, co získáme dostatečné množství deskriptorů, které nějakým způsobem přispívají k odlišení jednotlivých geometrických objektů, je potřeba od sebe objekty klasifikovat. Ke klasifikaci bylo v tomto případě využito neuronové sítě typu *patternnet* popsané výše v kapitole 1.4. Nejdříve jsme tedy v programovacím prostředí vygenerovali velké množství geometrických objektů, které obsahovali pouze hodnoty 1 a 0. Pro dokázání statistické úspěšnosti bylo pro každý tvar vygenerováno 200 objektů o různých parametrech. Každý vypočítaný deskriptor pro každý jednotlivý objekt byl zaznamenán jako příznak do matice k odpovídajícímu tvaru. Do matice jsou ukládány všechny vypočítané příznaky, tedy jak všechny geometrické, tak i

fourierovské. Poté, co je matice příznaků vytvořena, je využita jako vstupní parametr do neuronové sítě. Je vytvořena také anotace pořadí, ve kterém se objekty tvořili, abychom mohli následný výsledek klasifikace porovnat. Následně jsme také vhodně nastavili parametry trénování sítě. Například maximální počet epoch, parametr *useParallel* při trénování, který zrychlí celý proces neuronové sítě, abychom se dříve dostali k výsledku. Data jsme rozdělili na trénovací a testovací množinu vždy v poměru 80:20 a testování fungovalo jako pětinasobná křížová validace. Výstup sítě je poté převeden na jednořádkový vektor, který obsahuje seřazené hodnoty od 1 do 6 podle toho, jako který tvar síť klasifikovala objekt. Hodnota 1 v tomto případě odpovídala kruhu a hodnota 6 elipse. V našem případě jsme při porovnání mezi anotací a výsledkem sítě zjistili shodu 100%. To ovšem v případě, že nebyly obrazy žádným způsobem narušeny. K této správné klasifikaci jsme si vystačili s poměrně nízkým číslem deskriptorů. Větší množství vypočítaných deskriptorů už v tomto případě nemělo žádný efekt na úspěšnost klasifikace. Tedy byl tímto způsobem algoritmus dostatečně optimalizován.

Nyní se podíváme, jak by byl tento algoritmus účinný v případě, že obrazy obsahují šum. Pokud k vygenerovanému obrazu přičteme náhodný šum, můžeme na Obr. 2.11 vidět zašuměný obraz kruhu.



Obr. 2.11: Vygenerovaný kruh obsahující náhodný šum

V takto zašuměném obrazu by se nám špatně hledal obrys objektu a to by značně znehodnotilo následující klasifikaci. Proto na tento obraz aplikujeme gaussův filtr. Filtrovaný obraz pak můžeme vidět na Obr. 2.12.



Obr. 2.12: Filtrovaný obraz od náhodného šumu s pomocí gaussova filtru

Tento obraz je ovšem potřeba převést opět na binární. Toho jsme docílili v programovacím prostředí díky jednoduchému prahování. Při prahování filtrovaného obrazu se ovšem v některých případech stane, že se v obrazech u okrajů nachází malé objekty. Je to způsobené tím, že převádíme filtrovaný obraz, který předtím obsahoval šum. Tyto malé objekty by následnou klasifikaci znehodnotili nebo dokonce úplně přerušili. Těchto artefaktů jsme se zbavili s pomocí morfologické operace eroze, popsané v sekci 1.1.1. Po testování algoritmu se ovšem ve výjimečných případech v obrazu stále objevovali drobné objekty, a proto jsme obrazy ještě filtrovali mediánovým filtrem. Kromě šumu jsme také přidali ke každému vygenerovanému tvaru jako parametr náhodnou rotaci do  $180^\circ$ .

Nyní se opět vrátíme ke klasifikaci velkého množství náhodně vytvořených geometrických objektů naší neuronovou sítí. Kromě toho, že v tomto případě jsou obrazy kvůli rotaci, náhodnému šumu a jeho filtraci mírně náročnější pro klasifikaci, můžeme také mírně upravit vypočítané deskriptory, abychom dosáhli lepších výsledků klasifikace. Například přidáme fourierovský deskriptor, který byl popsán výše a to podíl rozptylu a střední hodnoty vzdálenosti obrysu od centroidu.

Pokud nyní otestujeme náš algoritmus na tyto vygenerované obrazy o stejném počtu jako předtím, tedy aby každý geometrický objekt byl vygenerován 200krát, můžeme pozorovat podobný výsledek jako u nezašuměných obrazů. V tomto případě nám shoda anotace a výsledku klasifikace neuronovou sítí vyšla v rozmezí 99,58-100%, tedy můžeme klasifikaci tvarů považovat za velmi úspěšnou i v tomto případě. Parametry nastavení neuronové sítě jsme neměnili. Nicméně výsledky se budou vždy lišit v řádech malých čísel z toho důvodu, že jsou všechny geometrické objekty generovány o náhodných rozměrech a záleží, jak jsme vybrali data do křížové validace. Vždy platí, že testovací data se nesmí při křížové validaci opakovat. V tomto případě je také náhodně generován šum, tedy nikdy nevíme, jakým způ-



sobem přesně tyto faktory úspěšnost klasifikace ovlivní. V tomto případě nebyla úspěšnost 100%, protože došlo k záměně kruhu a elipsy u jednoho případu, ke kterému mohlo dojít například kvůli způsobené deformaci původního obrazu při filtraci šumu a následném špatném zařazení na základě vypočítaných deskriptorů.

### 2.1.5 Diskuze výsledků klasifikace geometrických objektů

V této kapitole zhodnotíme zjištěné výsledky řešení této práce. Z výše uvedených částí praktické části jsme se mohli přesvědčit o tom, že algoritmy, kterými jsme testovali data, byly úspěšné.

V části předzpracování dat jsme se zabývali tím, že je potřeba, aby vstupní obrazy do algoritmu byly složeny z hodnot 1 a 0, abychom potlačili zbytečné detaily a skutečně dostali pouze obrys nějakého objektu, se kterým následně v tvarové analýze pracujeme. Generované geometrické objekty už byly takto přímo vytvořeny a stačilo tedy získat jejich obrys. Klíčové bylo poté získání kontur všech obrazů, ať už generovaných nebo načtených do programovacího prostředí. Na Obr. 2.4 jsme mohli pozorovat, že získání obrysu vygenerovaného geometrického objektu bylo úspěšné a stejným způsobem jsme získali i obrys dalších objektů. Stejně tak jsme mohli vidět na Obr. 2.5, že provedení obrysového trasování bylo také úspěšné, protože všechny obrysy objektů v obrazu byly zaznamenány a označeny červenou barvou.

V další části jsme se zabývali získáním tvarových deskriptorů tvarů popsaných v kapitole 1.3.4. Otestovali jsme na Obr. 2.6, že pokud je vygenerovaný objekt v obrazu skutečně kruh, vypočítaná míra kruhovitosti nám vyjde velmi podobná hodnotě 1. Následně jsme se také přesvědčili o tom, že pokud bychom měřili kruhovitost pro další tvary, jako je čtverec, obdélník a další, výsledná míra kruhovitosti by vycházela jinak, v tomto případě nižší než jedna. V Tab. 2.1 jsme poté mohli vidět tyto vypočítané hodnoty kruhovitosti všech generovaných geometrických objektů. Tímto způsobem jsme mohli pro všechny generované objekty získat soubor vlastností, které vypovídají o různých tvarových geometrických deskriptorech.

Následně jsme se také zabývali získáním fourierovských deskriptorů v sekci 2.1.3 a zobrazili jsme si výsledky vzdálenosti bodů obrysu od centroidu objektu v závislosti na úhlu pro několik různých generovaných objektů. Z ukázaných příkladů jsme mohli zhodnotit, že se tato charakteristika pro každý geometrický tvar velmi liší. Z grafů této charakteristiky můžeme pozorovat zejména píky, které odpovídají vrcholům objektů. Výjimkou je tedy v tomto případě kruh, který vrcholy neobsahuje a proto se jeho charakteristika jeví přibližně jako vodorovná úsečka, protože vzdálenost jeho obrysu od centroidu je přibližně konstantní. Je tedy zjevné, že fourierovské deskriptory významně přispějí k úspěšné klasifikaci. Všechny vypočítané

hodnoty deskriptorů jsou poté použity jako příznaky pro následnou klasifikaci neuronovou sítí.

Nejdříve jsme ještě otestovali detekci tvarů a jejich klasifikaci na základě minimální vzdálenosti od těžiště a dalších výpočtů s touto vzdáleností, kde algoritmus po správné detekci objektu do něj vypsal název odpovídajícího tvaru. Tento jednoduchý algoritmus fungoval na principu zadaných podmínek, podle kterých se rozhodoval dle zjištěných parametrů objektů. Podle toho, které hodnotě daného parametru se objekt blížil nejvíce, byl následně klasifikován. Tento jednoduchý algoritmus jsme otestovali na obrazu, který obsahoval více objektů najednou. Na Obr. 2.10 tedy můžeme vidět, že bylo správně klasifikováno několik tvarů. Algoritmus byl tedy v tomto případě úspěšný a to i pro vygenerované tvary přímo v programovacím prostředí.

Generování geometrických objektů jsme v programovacím prostředí nastavili tak, aby každý objekt byl vytvořen 200krát o různých parametrech. Pro čtverec, obdélník, trojúhelník to tedy znamenalo délky stran. U kosočtverce jsme také řešili úhel. U elipsy byly vždy nastaveny jiné hodnoty hlavní a vedlejší osy. U kruhu jsme zadávali jako parametr poloměr. Postupně jsou tedy v cyklu generovány tyto geometrické tvary o různých parametrech, kde jsme například některé postupně zvětšovali a některé se postupně zmenšovali. Pro každý vygenerovaný tvar byl získán obrys a bylo tedy vypočítáno několik tvarových deskriptorů (geometrických a fourierovských) a tyto vypočítané příznaky byly uloženy do matice. Po vytvoření matice příznaků jsou následně normalizovány a s pomocí dopředné neuronové sítě jsou klasifikovány tvary. V případě generovaných tvarů o náhodných parametrech bez jakéhokoliv rušení jsme vždy dostávali výsledek 100% úspěšné klasifikace.

Abychom ale otestovali úspěšnost algoritmu i při dalších podmínkách, tak byl vytvořen další test, kde byl do generovaných obrazů přičítán náhodný šum, který jsme následně filtrovali gaussovým filtrem. Po filtraci bylo ale potřeba převést obrazy opět na binární. Kromě šumu jsme tentokrát přidali ke každému vygenerovanému tvaru také náhodnou rotaci. Následně už algoritmus fungoval na stejném principu a tentokrát nás zajímalo, jaká bude jeho účinnost s těmito přidanými faktory. Nejdříve jsme testovali, jaké nastavení počtu skrytých vrstev a neuronů nám bude podávat nejlepší výsledky. V Tab. 2.2 tedy můžeme vidět nejnižší hodnotu klasifikace pro různá nastavení sítě.

n	Počet vrstev	Počet neuronů	Nejnižší shoda [%]
1.	1	8	99,17
2.	2	8, 6	99,17
3.	3	8, 6, 4	92,50
4.	1	10	99,17
5.	2	10, 8	99,58
6.	3	10, 8, 6	99,17
7.	1	12	99,17
8.	2	12, 10	99,17
9.	3	12, 10, 8	99,17

Tab. 2.2: Tabulka s nejnižší hodnotou klasifikace geometrických objektů pro různý počet vrstev a neuronů

Z Tab. 2.2 můžeme vidět, že nastavení sítě příliš neměnilo výsledek. Nejvyšší hodnota úspěšnosti byla u všech nastavení 100%. Nicméně nejlepší spodní hranice výsledku jsme dosáhli při použití dvou skrytých vrstev a počtu neuronů 10 a 8, a proto jsme nadále používali tuto síť. Výsledek klasifikace touto neuronovou sítí dle křížové validace nám tedy vyšel v rozmezí 99,58 - 100%. Pro další vypočítané hodnoty je ještě sestrojena následující Tab. 2.3.

Nejnižší hodnota	Nejvyšší hodnota	Průměr	Medián
99,58%	100%	99,67%	99,58%

Tab. 2.3: Tabulka s vypočítanými hodnotami klasifikace geometrických objektů

Pro případ, kdy jsme dosáhli nejnižšího výsledku 99,58% je vypracovaná následující Tab. 2.4, kde TP odpovídá skutečně pozitivním výsledkům, FP odpovídá falešně pozitivním výsledkům a FN falešně negativním.

Tvar	TP	FP	FN
Kruh	39	0	1
Čtverec	45	0	0
Obdélník	35	0	0
Kosočtverec	34	0	0
Trojúhelník	51	0	0
Elipsa	35	1	0

Tab. 2.4: Tabulka s výsledky klasifikace geometrických objektů

Na následující Tab. 2.5 můžeme vidět spočítanou hodnotu senzitivity a pozitivní prediktivní hodnoty (PPH) pro tento případ.

Senzitivita	Pozitivní prediktivní hodnota
0,9958	0,9958

Tab. 2.5: Tabulka s výsledky senzitivity a PPH klasifikace

Je patrné, že klasifikace byla v tomto případě velmi úspěšná. Z tabulky můžeme vidět, že hodnota senzitivity a pozitivní prediktivní hodnoty je stejná. Zmíněná záměna kruhu a elipsy byla pravděpodobně způsobena tím, že při úpravách zašuměného kruhu byly jeho výsledné rozměry ovlivněny a tedy se jevil jako elipsa. K tomu může také přispět například malý rozměr vytvořeného kruhu.

Jako poslední variantu jsme ještě testovali algoritmus pro případ, že se generovali zpočátku velmi malé rozměry objektů. Pokud byl takový obraz rušený šumem a následně filtrován, mohli jsme ztratit část informace, která by pomohla s následnou klasifikací. I v takovém případě jsme se při několika pokusech vždy dostali k hodnotě úspěšnosti téměř 100%. Zde se jednalo o záměnu trojúhelníku a elipsy. Tato chyba je opět pravděpodobně způsobena tím, že pokud jsou rozměry objektu velmi malé, může se ztratit velká část informace při filtraci šumu a dalších procesech přípravy obrazu.

Ve výsledku tedy můžeme říct, že klasifikace naším algoritmem byla velmi úspěšná jak v případě generování geometrických objektů bez rušení, tak i v případě, že obsahovali šum a byly náhodně rotovány. Pro tento výsledek jsme si vystačili pouze s těmito deskriptory:

- Obsah/obvod
- Délka hlavní osy/ délka vedlejší osy
- Kruhovitost
- Relativní zastoupení složek v dolní desetině spektra
- Rozptyl vzdálenosti obrysu od centroidu/střední hodnota vzdálenosti obrysu od centroidu

Výsledek klasifikace by dle mého názoru mohla zlepšit ještě implementace dalších deskriptorů, zvláště takových, které nebudou příliš ovlivněny předešlými úpravami obrazu potřebnými pro odstranění šumu. Klasifikaci také může velmi ovlivnit množství generovaných tvarů a tedy i velikost matice příznaků. Mohlo by tedy pomoci zvýšit ještě počet generovaných tvarů, ovšem u případu, kdy jsou tvary zatíženy šumem, bude algoritmus trvat o dost déle, než dojde k výsledku. Můžeme tedy pak řešit otázku optimalizace versus výsledek.

## 2.2 Aplikace tvarové analýzy na medicínská data

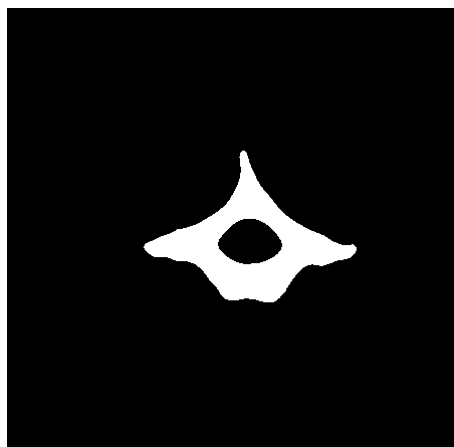
V této části se budeme zabývat použitím našeho algoritmu na reálná obrazová data. Doteď jsme pracovali převážně s obrazy vygenerovanými přímo v programovacím prostředí. Reálná obrazová data budou v tomto případě medicínská data dostupná z nemocničních nebo veřejných databází. Naším cílem bude klasifikace obratlů na thorakální (hrudní) a lumbální (bederní).

### 2.2.1 Obrazová CT data

Nadále budeme pracovat s databází deseti CT skenů páteře, které zároveň již obsahují segmentované thorakální a lumbální obratle. Data byla získána během běžné klinické práce od deseti pacientů. Každý sken obsahuje thorakální a lumbální část páteře. Datové soubory byly získány bez intravenózního kontrastu. Obrazová data byla přístupná ve formátu MetaImage, tedy hlavičkový soubor ve formátu mhd a data ve formátu raw.

Data jsou anotována tím způsobem, že jednotlivé obratle jsou označeny unikátním číslem. Každý sken obsahuje dvanáct thorakálních obratlů a pět bederních. Některé skeny obsahují také poslední krční obratle, které jsme ovšem pro další práci nepotřebovali.

Pro ukázkou našich dat a srovnání thorakálních a bederních obratlů můžeme na Obr. 2.13 vidět první thorakální obratel a na Obr. 2.14 první lumbální obratel.



Obr. 2.13: Jeden řez obratle Th1

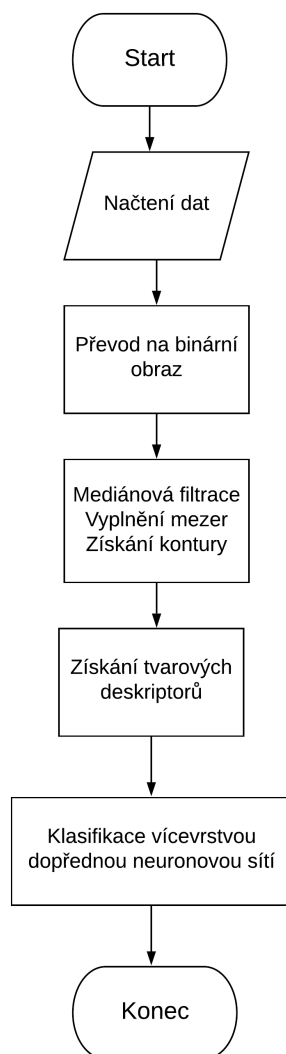


Obr. 2.14: Jeden řez obratle L1

Tato data dostupná z databáze jsme museli ovšem před otestováním našeho algoritmu vhodně upravit. Nejdříve jsme upravili formát dat, aby byla vhodná pro načtení do programovacího prostředí. Mezi další úpravy poté patřilo například zajištění toho, že hodnoty, které obraz obsahuje, budou po načtení odpovídat hodnotám obrazu v jeho původním formátu, než jsme ho konvertovali. To jsme zajistili odečtením vhodné konstanty od načtených obrazů. Také jsme vytvořili jeden soubor dat všech obratlů, abychom měli všechna data pohromadě a lépe se s nimi pracovalo. Na závěr bylo také tedy důležité z dat odstranit řezy obratlů, které nás pro klasifikaci nezajímají. Některé skeny totiž, jak bylo již výše psáno, obsahovali také krční obratle.

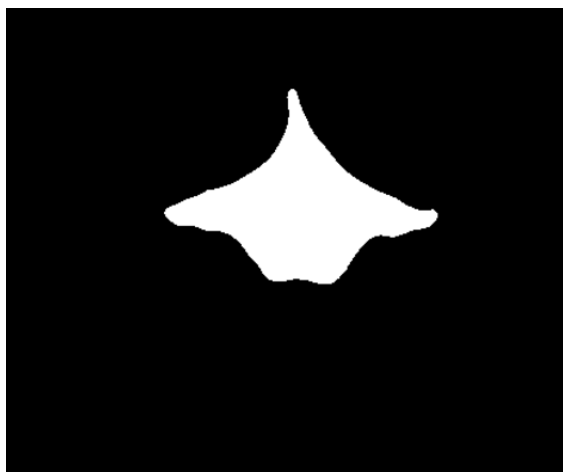
### 2.2.2 Klasifikace obratlů

Stejně jako v případě s generovanými daty bylo i zde nutné obrazy obsahující jednotlivé obratle vhodně upravit pro následující tvarovou analýzu. Vybrali jsme vždy pět řezů jednoho obratle tím způsobem, že jsme našli jeho prostřední řez, který obsahoval nejvíce informace a od něj jsme ukládali další řezy. Z vytvořeného souboru dat jsme vymazali nevhodné snímky obratlů, tedy například které obsahovali více objektů nebo které i po jejich vyplnění neměli vhodný tvar. Protože prvních dvanáct obratlů vždy odpovídá thorakálním, ale každého obratle jsme měli několik řezů, byla tentokrát vytvořena anotace podle unikátních hodnot obratlů, aby všechny obratle do hodnoty dvanáct odpovídaly thorakálním a zbytek lumbálním. Po těchto úpravách byl tedy vytvořen celkový soubor s 575 obrazy, na kterém jsme testovali náš algoritmus. Jeho princip je zobrazený na vývojovém diagramu na Obr. 2.15.



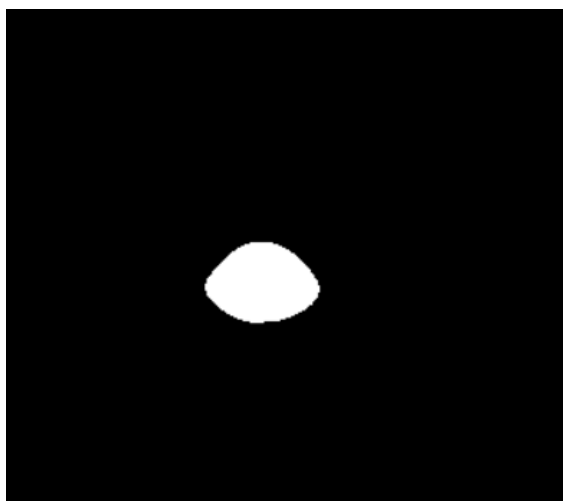
Obr. 2.15: Vývojový diagram algoritmu pro klasifikaci obratlů

Následně jsme vytvořili z každého obrazu obratle dva obrazy, kde první odpovídal celému tvaru obratle po zaplnění mezer a druhý páteřnímu kanálu. Jako příklad prvního obrazu uvedeme obratel Th1, který můžeme pozorovat na Obr. 2.16. Kromě toho byl tento obraz se zaplněným páteřním kanálem ještě filtrován mediánovým filtrem zvláště kvůli odstranění malých rušivých objektů, které obsahovali některé snímky.



Obr. 2.16: Vyplněný obratel Th1 po mediánové filtraci

Druhý obraz, který odpovídá pouze páteřnímu kanálu, je také z obratle Th1 a na následujícím Obr. 2.17 je zobrazený také po mediánové filtraci. Tento obraz jsme získali tak, že jsme od obrazu s vyplněným kanálem odečetli původní obraz obratle. V některých případech po odečtení obrazů neobsahoval výsledný obraz pouze páteřní kanál, ale také menší rušivý objekt. Zde bylo tedy nutné z vypočítaného souboru vlastností odstranit hodnoty vypočítané pro menší objekt nebo objekty než páteřní kanál, abychom dále pracovali pouze s ním.

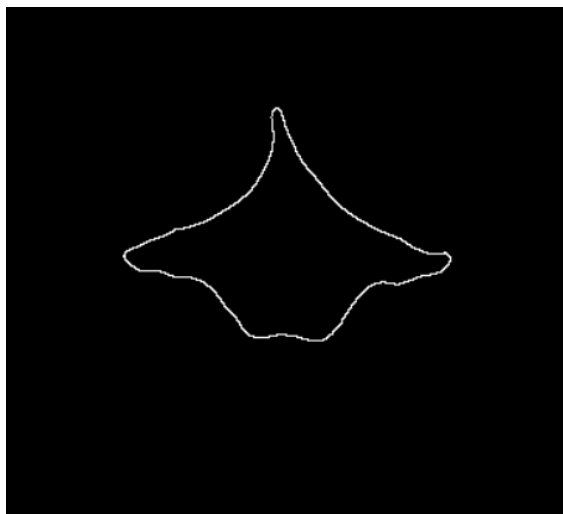


Obr. 2.17: Páteřní kanál Th1 po mediánové filtraci

Vyplnění páteřního kanálu bylo v tomto případě důležité zejména proto, abychom správně získali konturu obratle. Tentokrát jsme pro nalezení kontury použili v programovacím prostředí morfologickou operaci, která našla konečné body objektu.



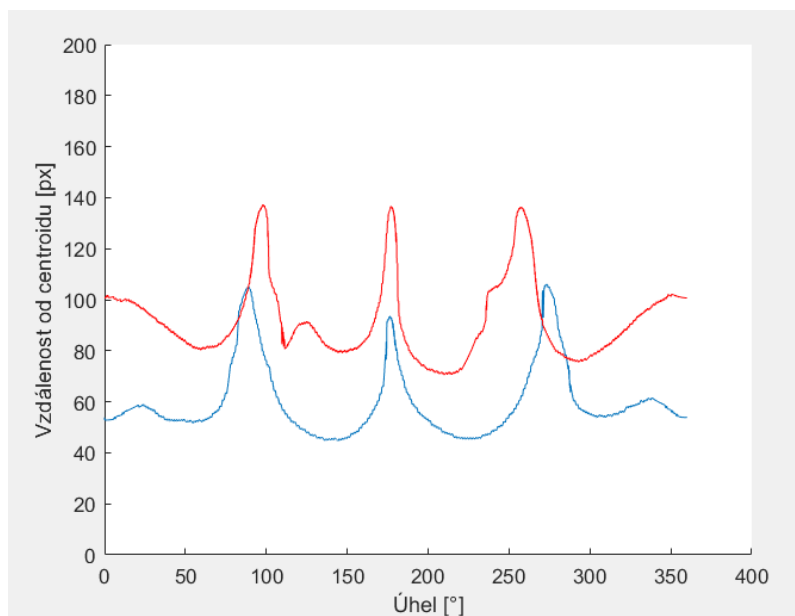
Na následujícím Obr. 2.18 můžeme pozorovat získanou konturu obratle, se kterou poté budeme později pracovat.



Obr. 2.18: Nalezená kontura obratle Th1 po předešlých úpravách

Stejným způsobem jsme mohli nalézt také konturu páteřního kanálu tohoto obratle Th1. Takto byly upraveny a nalezeny kontury i všech ostatních obrazů obratlů, nicméně pro ilustraci nám stačí tyto příklady. U každého obratle byly potom díky programovacímu prostředí získány jak geometrické, tak i fourierovské deskriptory a to jak pro vyplněný obratel, tak i pro páteřní kanál. Použili jsme stejné deskriptory jako v případě klasifikace generovaných geometrických objektů, ale také několik nových, které později uvedeme.

Některé z deskriptorů jsme aplikovali také na páteřní kanály. Zejména takové, které by mohly přispět k lepšímu rozlišení thorakálních a lumbálních obratlů. Využili jsme tedy zejména deskriptor *obsah/obvod* a fourierovské deskriptory na páteřní kanál. Nyní se podíváme na konkrétní příklad toho, jak se liší křivka vzdálenosti bodů obrysu od centroidu u thorakální obratle Th1 a lumbální obratle L5. Na Obr. 2.19 můžeme vidět, jak se tyto křivky liší.



Obr. 2.19: Vzdálenost bodů obrysu od centroidu u thorakálního obratle Th1 (modrou barvou) a lumbálního obratle L5 (červenou barvou)

Z výše uvedeného obrázku můžeme vyvodit, že i v případě klasifikace obratlů budou mít deskriptory založené na tomto principu dobrý vliv na úspěšnou klasifikaci.

### Výběr příznaků

Jak bylo výše zmíněno, abychom dosáhli úspěšné klasifikace, musíme vybrat vhodné deskriptory, které skutečně pomohou rozlišit dané objekty od sebe. Díky tomu získáme vhodné příznaky, které zvýší účinnost algoritmu. K tomu jsme využili korelační analýzu, popsanou v kap. 1.4.2. Vždy jsme si tedy uložili sadu vypočítaných parametrů, u které jsme se snažili zjistit vzájemnou korelaci příznaků. Ze získané matice příznaků jsme nejdříve spočítali matici korelačních koeficientů  $R$ , ve které jsme mohli pozorovat vzájemnou korelaci příznaků. Nicméně z této matice by bylo obtížné určit, které příznaky ponechat a které příznaky bychom měli odstranit. Tato matice byla proto ještě přepočítána dle vzorce (1.36) na matici  $t$ , která odpovídá testové statistice. Z této matice jsme už mohli provést  $t$ -test, u kterého jsme testovali následující hypotézy:

$$H_0 : r = 0 \quad (2.2)$$

$$H_1 : r \neq 0 \quad (2.3)$$

Sledujeme tedy, jestli korelace příznaků je nulová či není. Podle zadaných hypotéz je v případě nulové hypotézy korelace dvou příznaků nulová. Následně srovnáme

testovou statistiku  $t$  s kvantilem Studentova  $t$  rozdělení, které přísluší oboustranné alternativě hladiny významnosti  $\alpha = 0,05$ . Díky tomu získáme poslední matici, která nám dle zadaných hypotéz určuje, jestli je mezi danými příznaky korelace či nikoliv.

Samozřejmě vždy budeme muset volit kompromis mezi tím, jaké příznaky použijeme. Někdy totiž můžeme získat příznak, který není korelovaný se všemi ostatními a tedy bychom mohli zvažovat, jestli ho ponecháme. Nicméně může se například jednat o případ, kdy jsou s tímto daným příznakem korelované všechny ostatní a to nám pomůže se rozhodnout, že takový příznak vyřadíme.

Pokud se podíváme na Tab. 2.6, ve které hodnoty 1 odpovídají korelovaným příznakům a hodnoty 0 odpovídají nekorelovaným příznakům, můžeme se rozhodnout, které příznaky ponecháme a které vyloučíme, tedy vyloučíme odpovídající tvarový deskriptor.

Příznak	1.	2.	3.	4.	5.	6.	7.	8.	9.
1.	N	0	1	1	1	1	1	0	1
2.	0	N	1	1	1	1	0	0	1
3.	1	1	N	1	1	1	0	0	1
4.	1	1	1	N	1	1	0	0	1
5.	1	1	1	1	N	1	0	0	1
6.	1	1	1	1	1	N	1	1	1
7.	1	0	0	0	0	1	N	1	1
8.	0	0	0	0	0	1	1	N	1
9.	1	1	1	1	1	1	1	1	N

Tab. 2.6: Tabulka s výsledky korelační analýzy. Vyjadřuje korelaci mezi jednotlivými příznaky dle zadaných hypotéz.

V Tab. 2.6 můžeme vidět hodnoty 1 a 0 dle zadaných hypotéz, tedy jestli jsou příznaky mezi sebou korelované nebo nikoliv. Na diagonále jsou hodnoty, které nás nezajímají, protože vypovídají o korelaci příznaku sama se sebou. Potom už je na nás, jaký si nastavíme práh vyloučení příznaků. Podle výsledných hodnot v tabulce jsme v tomto případě vyřadili příznaky 6 a 9, které obsahují pouze hodnoty 1 korelace s ostatními příznaky. Nicméně je důležité si uvědomit, že pokud jsou příznaky korelované, neznamená to ještě, že k úspěšnosti klasifikaci žádným způsobem nepřispějí. Účinnost příznaků můžeme částečně kontrolovat také podle výsledku klasifikace neuronovou sítí.

Na tomto principu jsme tedy postupně vybírali příznaky, které zlepší výsledek klasifikace. Ze všech vypočítaných deskriptorů, které jsme ponechali dle výběru pří-

znaků, jsme opět vytvořili matici zastupující jednotlivé parametry. Hodnoty parametrů jsme tentokrát normalizovali pro selekci příznaků, a aby neuronová síť dosáhla lepších výsledků. Matici obsahující všechny vypočítané hodnoty deskriptorů z daných obratlů jsme použili jako vstup do neuronové sítě. Pro získání lepších výsledků klasifikace jsme tentokrát testovali neuronovou síť pro různý počet skrytých vrstev a neuronů. Trénování sítě většinou opět trvalo pouze v řádu 300 iterací. Soubor dat je rozdělen podle pětinasobné křížové validace na trénovací a testovací data, tedy v poměru 80:20. Výsledky mohli být při každé klasifikaci mírně odlišné, protože záleží na tom, jak jsme data rozdělili, nicméně při křížové validaci se nesmí testovací data nikdy opakovat. Při většině pokusů jsme dosáhli horní hranice úspěšnosti klasifikace téměř 100%, kterou jsme opět zjistili porovnáním naší vytvořené anotace pro testovací data obratlů s výstupem neuronové sítě.

### 2.2.3 Diskuze výsledků klasifikace obratlů

U klasifikace obratlů jsme tedy nejdříve museli zajistit, aby naše data obsahovala pouze ty obratle, které jsme od sebe chtěli rozlišit. Tímto nám po úpravách vzniklo 575 snímků. Podle toho, že první řezy byly vždy thorakálních obratlů a po nich následovali snímky lumbálních obratlů, jsme vytvořili anotaci. Poté jsme tedy z obrazu každé obratle vytvořili obrazy dva. První obsahoval obratel i s vyplněným kanálem a druhý obraz byl pouze páteřní kanál daného obratle. Tyto obrazy jsme vždy filtrovali mediánovým filtrem a následně jsme získali obrys jak obratle, tak páteřního kanálu s pomocí morfologické operace, která získala jejich konturu. V tomto případě jsme oproti testování algoritmu na geometrických objektech vždy získali dva soubory vlastností, ze kterých byly vypočítány deskriptory.

V Tab. 2.7 můžeme vidět výsledky testování algoritmu pro různý počet skrytých vrstev a skrytých neuronů v neuronové síti. V případě, že jsme použili více skrytých vrstev, byl počet neuronů nastaven tak, že byl v každé vrstvě jiný. Ve druhé vrstvě byl počet neuronů vždy nižší, což vedlo k lepším výsledkům. Testování tedy probíhalo jako pětinasobná křížová validace a byla vybrána nejnižší získaná shoda výstupu sítě s anotací.

n	Počet vrstev	Počet neuronů	Nejnižší shoda [%]
1.	1	8	92,17
2.	2	8, 6	92,17
3.	3	8, 6, 4	89,57
4.	1	10	87,83
5.	2	10, 8	93,04
6.	3	10, 8, 6	92,17
7.	1	12	91,30
8.	2	12, 10	91,30
9.	3	12, 10, 8	92,17

Tab. 2.7: Tabulka s nejnižší hodnotou klasifikace obratlů pro různý počet vrstev a neuronů

Ve většině případů byla horní hranice úspěšnosti téměř 100%. Z Tab. 2.7 plyne, že nejvhodnější nastavení sítě bude pro nás síť se dvěma skrytými vrstvami s počtem neuronů deset a osm. Nicméně výsledky se mohou při různých pokusech mírně měnit, protože také záleží, jak byla data rozdělena.

Po několika otestováních algoritmu jsme s pomocí výběru příznaků korelační analýzou přímo v programovacím prostředí zjistili, že nejlepšího výsledku klasifikace s pomocí neuronové sítě jsme dosáhli při aplikaci těchto osmi deskriptorů:

- Obsah/obvod
- Délka hlavní osy/ délka vedlejší osy
- Kompaktnost
- Obsah/obvod páteřního kanálu
- Relativní zastoupení složek v dolní desetinně spektra
- Rozptyl vzdálenosti obrysu od centroidu/střední hodnota vzdálenosti obrysu od centroidu
- Relativní zastoupení složek v dolní desetinně spektra páteřního kanálu
- Střední frekvence vzdálenosti obrysu od centroidu u páteřního kanálu

Parametry neuronové sítě jsme tedy mírně upravili oproti předešlé klasifikaci s geometrickými objekty. Počet skrytých vrstev a neuronů byl tentokrát nastaven na výše zmíněné hodnoty, maximální počet epoch byl tentokrát 300, protože jich probíhal vždy nízký počet. Výsledek klasifikace touto neuronovou sítí dle křížové validace nám v jednom případě vyšel v rozmezí 93,04 - 98,26%. Pro tento případ je sestrojena následující Tab. 2.8.

Nejnižší hodnota	Nejvyšší hodnota	Průměr	Medián
93,04%	99,13%	96,52%	97,39%

Tab. 2.8: Tabulka s vypočítanými hodnotami klasifikace obratlů

Pro výsledek úspěšnosti 93,04% byla ještě vytvořena následující Tab. 2.9, kde TP odpovídá skutečně pozitivním výsledkům, FP odpovídá falešně pozitivním výsledkům a FN falešně negativním.

Obratel	TP	FP	FN
Thorakální	75	4	4
Lumbální	32	4	4

Tab. 2.9: Tabulka s výsledky klasifikace obratlů

V tomto případě můžeme z Tab. 2.9 vidět, že došlo k osmi špatným klasifikacím obratlů. Ve čtyřech případech se jednalo o to, že síť klasifikovala thorakální obratel jako lumbální a ve čtyřech případech se jednalo o to, že síť označila lumbální obratle jako thorakální. V Tab. 2.10 pak můžeme pozorovat další výpočty pro senzitivitu a pozitivní prediktivní hodnotu (PPH) testu.

Senzitivita	Pozitivní prediktivní hodnota
0,9304	0,9304

Tab. 2.10: Tabulka s výsledky senzitivity a PPH klasifikace obratlů

Z Tab. 2.10 můžeme stejně jako u klasifikace geometrických objektů vidět, že hodnota senzitivity a PPH jsou stejné, pokud počítáme výsledek pro celý test. Je to z toho důvodu, že máme stejný počet falešně pozitivních (FP) a falešně negativních (FN) detekcí. Tato hodnota také po vynásobení číslem sto odpovídá úspěšnosti klasifikace v procentech.

Z výše uvedených výsledků tedy můžeme usoudit, že i v případě klasifikace obratlů byl algoritmus velmi úspěšný. Dalšího zlepšení výsledku by se dalo dle mého názoru dosáhnout například ještě aplikací dalších deskriptorů, ovšem bylo by záhodné opět zkontrolovat, jestli přispívají k lepšímu výsledku. Dalším faktorem, který by mohl klasifikaci pomoci, by bylo ještě rozšířit databázi snímků, aby se neuronová síť mohla naučit na více datech a výsledek klasifikace by se tím mohl zlepšit. Nicméně v tomto případě jsme se snažili použít taková data, která mají smysluplný význam pro klasifikaci a další snímky by nemuseli nést dostatečnou informaci pro rozlišení obratlů.

### 3 Závěr

V této práci jsme se věnovali teorii tvarové analýzy, jejím postupům, metodám a její následné aplikaci. V teoretické části byly popsány základní operace zpracování obrazu jako je morfologické zpracování, segmentace a reprezentace obrazu. Věnovali jsme se několika příkladům tvarových deskriptorů, které mohou pomoci s klasifikací tvarů. Také jsme uvedli použitou klasifikační metodu a několik příkladů aplikace tvarové analýzy v praxi, které přímo nesouvisí s naší prací, ale poukazují na jiné využití a rozšíření metody tvarové analýzy.

Některé postupy popsané v teorii jsme poté implementovali v programovacím prostředí MATLAB. Algoritmus byl testován na vygenerovaných 2D obrazech obsahujících známé tvary, jako je kruh, čtverec, trojúhelník a další, na kterých bylo možné ověřit efektivitu metod. Použité metody v řešení tedy vycházeli z popsané teorie a z tvarových deskriptorů byly využity zejména geometrické a fourierovské. Hodnoty získané s pomocí deskriptorů nám poté pomohli klasifikovat různé tvary objektů. Díky tomu jsme byli nakonec schopni správně klasifikovat několik geometrických tvarů od sebe. Ke klasifikaci byla využita vícevrstvá dopředná neuronová síť.

Naše řešení se skládalo z několika základních částí. Nejdříve bylo potřeba vygenerovat nebo jinak získat vstupní obrazová data a následně je upravit pro další práci. Hlavní úpravou bylo vždy zajištění toho, aby byl vstupní obraz binární, což je pro tvarovou analýzu velmi podstatné. Díky tomu jsme mohli například ověřit metody popsané v segmentaci obrazu, mezi které patřilo obrysové trasování nebo z morfologických operací získání kontury. Poté jsme aplikovali některé deskriptory vhodné pro získání vlastností objektů nutných pro následnou klasifikaci tvarů. Protože jsme testovali algoritmus na velkém množství vstupních obrazů, můžeme výsledek klasifikace považovat za velmi úspěšný. V případě, kdy byl do vstupních obrazů přidán náhodný šum a také rotace, se klasifikátor opět projevil jako schopný rozpoznat tvary správně.

V další části, kde jsme pracovali s medicínskými obrazovými daty, bylo potřeba algoritmus mírně upravit, aby byl výsledek klasifikace opět úspěšný. Cílem bylo od sebe klasifikovat thorakální a lumbální obratle. Nejdříve byly tedy vybrány pouze řezy, se kterými jsme měli pracovat a které byly pro klasifikaci vhodné. Také se prokázalo jako vhodné, použít i další tvarové deskriptory, které jsme vybírali s pomocí korelační analýzy, pro lepší výsledek. Podle testů neuronové sítě se prokázalo jako vhodné použít síť se dvěma skrytými vrstvami. Zjistili jsme, že výsledek klasifikace obratlů byl velmi úspěšný, a tedy se nám podařilo dosáhnout cíle této práce.

# Seznam zkratek

1. 1-D - jednodimensionální (onedimensional)
2. CAD - s pomocí počítače (computer-aided)
3. CCH - histogram řetězového kódu (chain code histogram)
4. CT - výpočetní tomografie (computed tomography)
5. FD - fourierův deskriptor (fourier descriptor)
6. FN - falešně negativní (false negative)
7. FP - falešně pozitivní (false positive)
8. FT - fourierova transformace (fourier transformation)
9. GGO - ground-glass opacity
10. IF - index formy (form index)
11. MBR - minimum ohraničující obdélník (minimum bounding rectangle)
12. MRA - multiresoluční analýza (multiresolution analysis)
13. MSD - střední kvadratická vzdálenost (mean squared deviation)
14. NCCH - normalizovaný histogram řetězového kódu (normalized chain code histogram)
15. PPH - pozitivní prediktivní hodnota
16. SPHARM - sférické harmonické (spherical harmonics)
17. TP - skutečně pozitivní (true positive)
18. WD - vlnkové deskriptory (wavelet descriptors)
19. WT - vlnková transformace (wavelet transformation)



# Literatura

- [1] Al-Rousan, T.; Masad, E.; Tutumluer, E.; aj.: Evaluation of image analysis techniques for quantifying aggregate shape characteristics. *Construction and Building Materials*, vol. 21, nr. 5, 2007: s. 978–990, ISSN 09500618, doi:10.1016/j.conbuildmat.2006.03.005.
- [2] Arbeláez, P.; Maire, M.; Fowlkes, C.; aj.: Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, nr. 5, May 2011: s. 898–916, ISSN 1939-3539, doi: 10.1109/TPAMI.2010.161.
- [3] Bird, J. L.; Eppler, D. T.; Checkley Jr., D. M.: Comparisons of Herring Otoliths Using Fourier Series Shape Analysis. *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 43, nr. 6, 1986: s. 1228–1234, doi:10.1139/f86-152.
- [4] Bowman, E. T.; Soga, K.; Drummond, W.: Particle shape characterisation using Fourier descriptor analysis. *Géotechnique*, vol. 51, nr. 6, 2001: s. 545–554, doi: 10.1680/geot.2001.51.6.545.
- [5] Britannica, E.: Student's t-test | Definition, Formula, Example. <https://www.britannica.com/science/Students-t-test>, 2020, [cit. 2020-05-31].
- [6] Correlation: Statistics Solutions. Statistics Solutions: Dissertation and Research Consulting For Statistical Analysis. <https://www.statisticssolutions.com/correlation-pearson-kendall-spearman/>, 2020, [cit. 2020-05-31].
- [7] Costa, V.: Vision Methods to Find Uniqueness Within a Class of Objects. 04 2019, doi:10.13140/RG.2.2.13750.14401.
- [8] CRAMPTON, J. S.: Elliptic Fourier shape analysis of fossil bivalves: some practical considerations. *Lethaia*, vol. 28, nr. 2, 1995: s. 179–186, doi:10.1111/j.1502-3931.1995.tb01611.x.
- [9] Gerig, G.; Styner, M.; Jones, D.; aj.: Shape analysis of brain ventricles using SPHARM. In *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA 2001)*, Dec 2001, s. 171–178, doi:10.1109/MMBIA.2001.991731.
- [10] Gonzalez, R. C.; Woods, R. E.: *Digital image processing*. Upper Saddle River: Pearson, 3. edice, c2008, ISBN 978-0131687288.

- [11] Jalalian, A.; Mashohor, S.; Mahmud, R.; aj.: Foundation and methodologies in computer-aided diagnosis systems for breast cancer detection. *EXCLI journal*, vol. 16, 2017: str. 113, doi:10.17179/excli2016-701.
- [12] Jan, J.: *Medical Image Processing, Reconstruction and Analysis*. CRC Press, 2019-8-30, ISBN 9781315143699, doi:10.1201/b22391.
- [13] Kaganami, H. G.; Beiji, Z.: Region-Based Segmentation versus Edge Detection. In *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Sep. 2009, s. 1217–1221, doi:10.1109/IIH-MSP.2009.13.
- [14] Liang Shen; Rangayyan, R. M.; Desautels, J. E. L.: Application of shape analysis to mammographic calcifications. *IEEE Transactions on Medical Imaging*, vol. 13, nr. 2, June 1994: s. 263–274, ISSN 1558-254X, doi:10.1109/42.293919.
- [15] MathWorks: Pattern recognition network - MATLAB patternnet. <https://www.mathworks.com/help/deeplearning/ref/patternnet.html;jsessionid=9034fd5de9a2ed2481e39d5eba86>, 1994, [cit. 2020-05-31].
- [16] Mendonça, P. R. S.; Bhotika, R.; Sirohey, S. A.; aj.: Model-Based Analysis of Local Shape for Lesion Detection in CT Scans. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005*, Springer Berlin Heidelberg, 2005, ISBN 978-3-540-32094-4, s. 688–695.
- [17] Nabout, A. A.: Object Shape Recognition Using Wavelet Descriptors. *Journal of Engineering*, vol. 2013, 2013: s. 1–15, ISSN 2314-4904, doi:10.1155/2013/435628.
- [18] Šnábl, I.: Matematická Biologie Učebnice: Test Hypotézy O Nulové Korelaci Dvou Náhodných Veličin. <https://portal.matematickabiologie.cz/index.php?pg=aplikovana-analyza-klinickych-a-biologickych-dat>, 2020, [cit. 2020-05-31].
- [19] Rosin, P. L.: Measuring shape. *Machine Vision and Applications*, vol. 14, nr. 3, 2003: s. 172–184, ISSN 0932-8092, doi:10.1007/s00138-002-0118-6.
- [20] Styner, M.; Lieberman, J. A.; Gerig, G.: Boundary and Medial Shape Analysis of the Hippocampus in Schizophrenia. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003*, Springer Berlin Heidelberg, 2003, ISBN 978-3-540-39903-2, s. 464–471.

- [21] Svozil, D.; Kvasnicka, V.; Pospíchal, J.: Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, vol. 39, 11 1997: s. 43–62, doi:10.1016/S0169-7439(97)00061-0.
- [22] Thapar, S.; Garg, S.: Study and implementation of various morphology based image contrast enhancement techniques. *International Journal of Computing and Business Research*, 2012: s. 2229–6166, ISSN 2229-6166.
- [23] Toppr: Area and Perimeter: Definition, Formulae, Videos and Solved Examples. <https://www.toppr.com/guides/maths/mensuration/area-and-perimeter/>, [cit. 2020-05-31].
- [24] Udrea, M. R.; Vizireanu, N. D.: Iterative generalization of morphological skeleton. *Journal of Electronic Imaging*, vol. 16, nr. 1, 2007: s. 1 – 3, doi: 10.1117/1.2713739.
- [25] Veeraraghavan A; Roy-Chowdhury, A. K.; Chellappa, R.: Matching shape sequences in video with applications in human movement analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, nr. 12, Dec 2005: s. 1896–1909, ISSN 1939-3539, doi:10.1109/TPAMI.2005.246.
- [26] Ye\*, X.; Lin, X.; Dehmeshki, J.; aj.: Shape-Based Computer-Aided Detection of Lung Nodules in Thoracic CT Images. *IEEE Transactions on Biomedical Engineering*, vol. 56, nr. 7, July 2009: s. 1810–1820, ISSN 1558-2531, doi:10.1109/TBME.2009.2017027.
- [27] Zhang, D.; Lu, G.: Shape-based image retrieval using generic Fourier descriptor. *Signal Processing: Image Communication*, vol. 17, 11 2002: s. 825–848, doi: 10.1016/S0923-5965(02)00084-X.
- [28] Zhang, D.; Lu, G.: Review of shape representation and description techniques. *Pattern Recognition*, vol. 37, nr. 1, 2004: s. 1–19, ISSN 00313203, doi:10.1016/j.patcog.2003.07.008.